Mundo AMSTRAD

LOS MEJORES LISTADOS PARA TU MICRO

EDITA 

-añoI-Nº 2

PVP 200 Pts. IVA Incl.

LISTADOS

ARCHIVO NUMERICO

GRANADAS

CEFALUS

ORGANO ELECTRONICO

MORTERO

GENERADOR GRAFICOS

KINETIK

AMSTRAD CPC

SENTINEL

FIREBIRD

TRANSMUTER

AMSTRAD SCHNEIDER

Code Masters

By Ian Richards

PVP CANARIAS 200 Pts.


¡Tu cinta para grabar guay!
HOLIDAY
HOLIDAY C-40
HOLIDAY
C-90
HOLIDAY
Tu cinta virgen
de 40, 60 y 90
HOLIDAY
C-90
HOLIDAY
V. CARRERA
Fabricada por IBEROFON, S.A.
Avenida de Fuentemar, 35 - Polígono Industrial de Coslada - MADRID
Teléfonos 671 22 00-04-08-12 - Télex 42797 FONO E - Telefax (91) 671 39 09
S ON S
Direct Metal DMM Mastering
COMPACT disc DIGITAL AUDIO
discoflex®



Director:
Angel Herrero Fernández
Director Técnico:
Luis Sanguino
Coordinador Editorial:
Félix Santamaría
Software:
Angel García
Secretaria Redacción:
Mercedes Matons
Diseño e Ilustraciones:
Javier Caballero
Colaboradores:
Antonio García
A. Gustavo Chico
Andrés M. García

Mundo AMSTRAD es una publicación del Grupo Editorial SYGRAN, S.A. Polig. Ind. Valdonaire. C/Apolonio Hernández. HUMANES (Madrid).
Director Adjunto:
Andrés Franco

Publicidad y suscripciones:
GENESIS
Tomás López, 3-6°
28009 Madrid
Tel. 401 77 54

Fotocomposición:
Speed Letra, S.A.
Imprime:
Gráficas Osiris, S.A.
Brañuelas, 29
Fuenlabrada
Distribuye:
G.M.E. Pza. Castilla, 3.
Madrid
Depósito Legal:
M-31674-1987

SUMARIO

LISTADOS

5 **Cefalus**
10 **Generador de caracteres gráficos**
13 **Organo electrónico**
22 **Granadas**
25 **Recogida**
28 **Mortero**
31 **Archivo numérico**

JUEGOS

16 **Transmuter**
17 **Druid**
18 **Toys**
19 **Sentinel**
20 **Game Over**
21 **Kinetik**



Os ofrecemos un truco con el que podeís realizar "VIDEO INVERSO" de manera casi instantánea. Con sólo hacer una llamada a la ROM de vuestro AMSTRAD el sistema operativo se encarga de invertir los colores de tinta y fondo con que aparecen los textos en la pantalla del monitor.

Valga éste ejemplo como demostración:

```
1 '******************
2 '* PROGRAMA PARA   *
3 '* INVERTIR TEXTO *
4 '* EN LA PANTALLA *
5 '******************
6 '
10 CALL &BB9C
20 PRINT "HOLA"
25 I=I+1:IF I=5 THEN 40
30 GOTO 20
40 I=0:GOTO 10
```

Con éste pequeño truco podrás proteger tus programas contra cualquier curioso que intente listarlo para saber cómo está hecho o para copiarlo. La llamada CALL &BB45 al sistema operativo se encarga, de un modo "drástico", de evitar que un programa pueda ser detenido durante su ejecución.

Teclea el siguiente ejemplo y podrás comprobarlo:

```
10 ' ************************
20 ' * PROTEGER UN PROGRAMA *
30 ' ************************
40 '
50 CALL &BB45
60 MODE 1
70 PRINT "PULSE <ESC> PARA INTERTAR"
80 PRINT "   DETENER EL PROGRAMA    "
90 PRINT
100 PRINT " SIGO FUNCIONANDO ";
110 GOTO 100
```

INDESCOMP PASA A MANOS DE AMSTRAD



Durante la celebración del 25 certamen de SONIMAG en Barcelona en los últimos días del mes de septiembre, fue dada a conocer en rueda de prensa a los medios informativos la noticia de la compra del 100% de las acciones del GRUPO INDESCOMP por parte de la firma inglesa AMSTRAD PLC de cuyos productos, INDESCOMP es distribuidor exclusivo en nuestro país.

Alan Sugar, presidente de AMSTRAD PLC, se desplazó hasta Barcelona para dar a conocer la noticia de la operación que supone la absorción por parte de la compañía inglesa del total de la plantilla de Indescomp.

Alan Sugar informó de su propósito de que en un plazo breve fueran construídos en España algunos de los componentes de los equipos AMSTRAD, lo cual daría un fuerte impulso a la empresa.

El acuerdo, pendiente de aprobación por parte del organismo competente del gobierno español, será primado en breve y el coste total de la operación podría alcanzar los 250 millones de libras esterlinas: 216 a la firma del contrato y el resto, en función de los beneficios obtenidos en los primeros 5 años.



PROGRAMACION AVANZADA DEL AMSTRAD

Anaya Multimedia

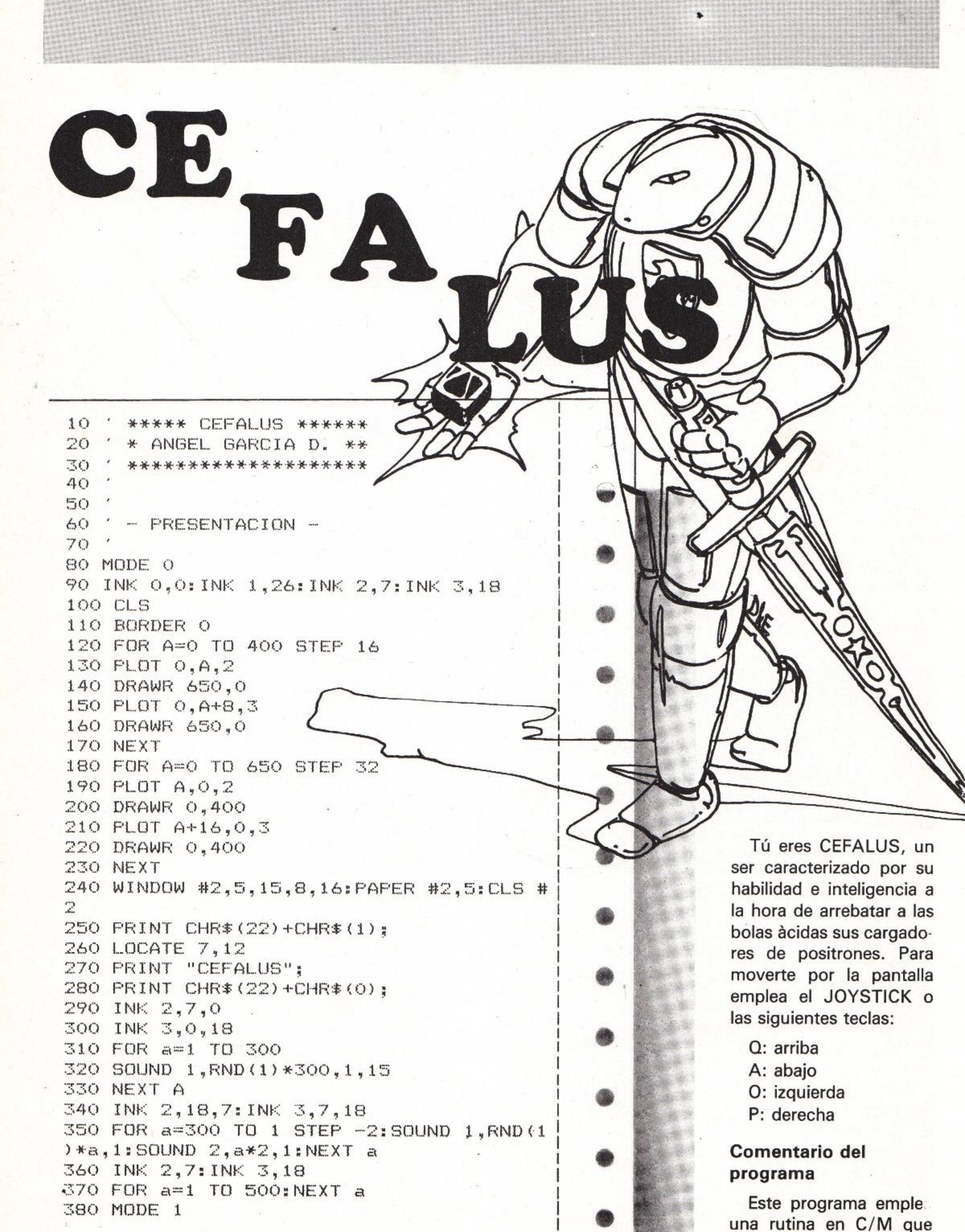


Una de las cualidades más importantes del Amstrad CPC 464 es la facilidad de acceso a las rutinas más importantes de la ROM, sin necesidad de ser un programador experimentado.

Don Thomasson analiza esta característica en los primeros capítulos del libro de forma clara y bien estructurada. Diseño de pantallas, entradas y salidas de cassette, etc., pueden ser reducidas a simples llamadas de subrutinas.

Otra de las cualidades más interesantes del Amstrad es su capacidad para comunicarse con el mundo exterior y la opción de ROM externa que permiten aumentar la capacidad del ordenador. En el libro se habla de todo esto con detalle, de cara a mostrar como aprovechar mejor todas las posibilidades que nos ofrece el Amstrad CPC 464.

CEFALUS



```
10 ' ***** CEFALUS ******
20 ' * ANGEL GARCIA D. **
30 ' ********************
40 '
50 '
60 ' - PRESENTACION -
70 '
80 MODE 0
90 INK 0,0:INK 1,26:INK 2,7:INK 3,18
100 CLS
110 BORDER 0
120 FOR A=0 TO 400 STEP 16
130 PLOT 0,A,2
140 DRAWR 650,0
150 PLOT 0,A+8,3
160 DRAWR 650,0
170 NEXT
180 FOR A=0 TO 650 STEP 32
190 PLOT A,0,2
200 DRAWR 0,400
210 PLOT A+16,0,3
220 DRAWR 0,400
230 NEXT
240 WINDOW #2,5,15,8,16:PAPER #2,5:CLS #
2
250 PRINT CHR$(22)+CHR$(1);
260 LOCATE 7,12
270 PRINT "CEFALUS";
280 PRINT CHR$(22)+CHR$(0);
290 INK 2,7,0
300 INK 3,0,18
310 FOR a=1 TO 300
320 SOUND 1,RND(1)*300,1,15
330 NEXT A
340 INK 2,18,7:INK 3,7,18
350 FOR a=300 TO 1 STEP -2:SOUND 1,RND(1
)*a,1:SOUND 2,a*2,1:NEXT a
360 INK 2,7:INK 3,18
370 FOR a=1 TO 500:NEXT a
380 MODE 1
```

Tú eres CEFALUS, un ser caracterizado por su habilidad e inteligencia a la hora de arrebatar a las bolas àcidas sus cargadores de positrones. Para moverte por la pantalla emplea el JOYSTICK o las siguientes teclas:

Q: arriba
A: abajo
O: izquierda
P: derecha

Comentario del programa

Este programa emple una rutina en C/M que

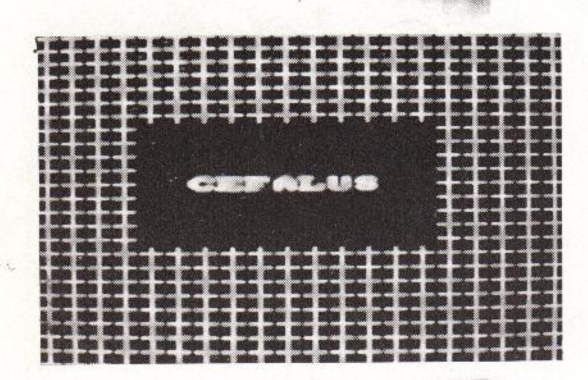
Cefalus

gestiona el movimiento de todas las figuras de la pantalla; las 6 bolas enemigas; nuestro muñeco y el bloque de energía, con animación de 2 gráficos.

10-390. Pantalla de presentación. Hecha en MODO 0. Se dibuja un entramado. Se abre ua ventana en el interior y se imprime el título. Se hacen algunos sonidos y se pasa a las instrucciones.

400-470. Subrutina que imprime el texto.

480-590. Se imprimen las instrucciones y se espera la pulsación de una tecla para continuar.



Antes de imprimir este mensaje son llamadas dos subrutinas encargadas de redefinir los gráficos que se van a usar. La segunda introduce el C/M en memoria. El método usado es meter los códigos en hexadecimal seguidos en una cadena de caracteres, que luego se fragmenta de dos en dos elementos y se introducen en memoria. La letra Z es la marca de final.

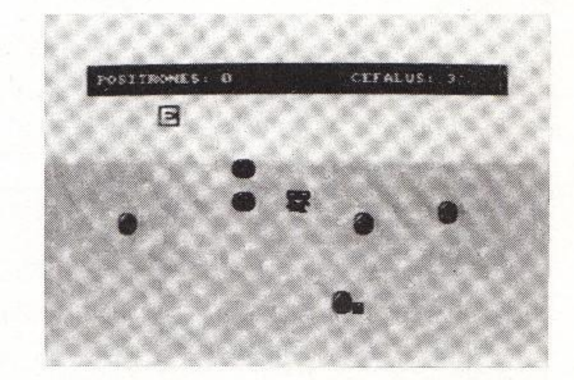
```
390 GOTO 480
400 '
410 ' - IMPRIME TEXTO -
420 '
430 PRINT TAB(20-LEN(a$)/2):FOR a=1 TO L
EN(a$)
440 b$=MID$(a$,a,1):PRINT b$;:IF b$<>" "
 THEN SOUND 1,A,1,15
450 NEXT a
460 PRINT:PRINT:FOR a=1 TO 200:NEXT a:RE
TURN
470 '
480 ' - INSTRUCCIONES -
490 '
500 a$="<<< C E F A L U S >>>":GOSUB 430
510 PRINT:A$="AYUDA A CEFALUS A RECOJER"
:GOSUB 430
520 A$="LOS CONTENEDORTES DE ENERGIA":GO
SUB 430
530 A$="PARA AUMENTAR SU CARGADOR DE POS
ITRONES.":GOSUB 430
540 A$="PERO TEN CUIDADO CON LAS BOLAS A
CIDAS":GOSUB 430
550 PRINT:A$="TECLADO (Q.A.O.P) O JOYSTI
CK":GOSUB 430
560 GOSUB 1310:GOSUB 1240
570 PRINT:A$="PULSA UNA TECLA PARA EMPEZ
AR":GOSUB 430
580 A$=INKEY$:IF INKEY$="" THEN 580
590 '
600 ' - INICIALIZA VARIABLES -
610 '
620 V=3
630 MODE 1
640 INK 0,26:INK 1,0:INK 2,15:INK 3,2
650 PAPER 0:BORDER 26
660 CLS
670 WINDOW 1,40,4,25:PEN 1:CLS
680 WINDOW #1,1,40,1,3:PAPER#1,2:PEN#1,0
:CLS#1
690 '
700 ' - MOVIMIENTO MALOS -
710 '
720 GOSUB 1190
730 RESTORE 1070:D=40000
740 READ A:POKE D,A:D=D+1
750 IF A=255 THEN GOTO 770
760 GOTO 740
770 DEFINT X,Y:X=10:Y=20
780 H=1+INT(RND*19):I=1+INT(RND*39):POKE
 40015,H:POKE 40016,I:FOR B=100 TO 200 S
TEP 10:SOUND 2,B,1,15:NEXT B
790 GOSUB 1190
800 '
810 ' - BUCLE PRINCIPAL -
```

```
820 '
830 CALL 32768
840 IF PEEK(39999)=1 THEN GOSUB 970:GOTO
 630
850 IF INKEY(67)=0 OR JOY(0)=1 THEN X=X-
1:IF X<1 THEN X=1
860 IF INKEY(69)=0 OR JOY(0)=2 THEN X=X+
1:IF X>20 THEN X=20
870 IF INKEY(34)=0 OR JOY(0)=4 THEN Y=Y-
1:IF Y<1 THEN Y=1
880 IF INKEY(27)=0 OR JOY(0)=8 THEN Y=Y+
1:IF Y>39 THEN Y=39
890 POKE 40003,X:POKE 40004,Y
900 IF X=H AND Y=I THEN P=P+15:GOTO 780
910 IF X+1=H AND Y+1=I THEN P=P+15:GOTO
780
920 GOTO 830
930 '
940 ' - PIERDE VIDA -
950 '
960 DATA 1,10,10,2,1,20,2,1,15,1,172,176
970 POKE 40010,180:POKE 40011,184
980 FOR A=0 TO 100 STEP 10:SOUND 129,RND
(1)*300,20,15,0,0,31:CALL 32768:NEXT A
990 V=V-1: IF V>=0 THEN RETURN
1000 FOR a=0 TO 40:SOUND 1,A*10,10,15:A$
=INKEY$:NEXT a
1010 PEN 1:LOCATE 16,11:PRINT"GAME OVER"
1020 IF INKEY$="" THEN 1020
1030 RUN 620
1040 '
1050 ' - POSICIONES INICIALES -
1060 '
1070 DATA 3,10,20,10,20,0,0,0,0,1,160,16
4
1080 DATA 3,10,10,10,10,0,0,0,0,2,168,16
8
1090 DATA 1,12,12,1,2,20,0,1,39,2,172,17
6
1100 DATA 1,15,15,0,2,20,1,2,39,3,176,17
2
1110 DATA 1,10,10,2,2,20,2,2,39,2,172,17
6
1120 DATA 1,10,25,2,3,18,0,5,35,3,176,17
2
1130 DATA 1,5,30,0,1,20,2,1,39,2,172,176
1140 DATA 1,5,10,2,1,20,2,1,39,2,172,176
1150 DATA 255
1160 '
1170 ' - MARCADORES -
1180 '
1190 LOCATE#1,2,2:PRINT#1,"POSITRONES:";
P;TAB(26);"CEFALUS:";V
1200 RETURN
1210 '
```

Cuando introduzcas la zona de los DATAS ten mucho cuidado, pues un error aquí podría resultar fatal.

600-690. Se inicializan las variables. Vidas = 3. Modo de pantalla el 1. Se definen dos ventanas; una para el juego y otra para los marcadores.

700-800. Se imprimen los marcadores. Y los datos de los movimientos de los malos son introducidos en la memoria. Los



DATAS leídos son los de las líneas 1070-1150. La forma en que se codifica es:

Primer byte: 3. Indica que es un objeto en cuyo movimiento va a ser controlado desde BASIC y

los datos son interpretados de la siguiente forma:

3, X, Y, XB, YB, 0, 0, 0, 0, Color, Animación 1, Animación 2. XB e YB son las coordenadas de borrado.

Cefalus

1. Gráfico movido por la rutina en Código máquina. Se comprueba además si se choca con el gráfico 0 y se indica en la posición de memoria 39999. Los datos son interpretados de la siguiente manera:

1, X, Y, incrementoX, Tope X, Tope X', Incremento Y, Tope Y, Tope Y', color, A1, A2.

255. Marca de final de datos.

810-920. Bucle principal del juego. Se llama en primer lugar a la rutina de movimiento. Luego se inspecciona el byte de ohoque, que si es 1, hemos chocado.

Luego se lee el teclado y el joystick para modificar las coordenadas de nuestro muñeco si fuera necesario.

Se POKEan las coordenadas.

Se comprueba si cogemos la energía. Si la cogemos se cambia de sitio y se suman 15 puntos.

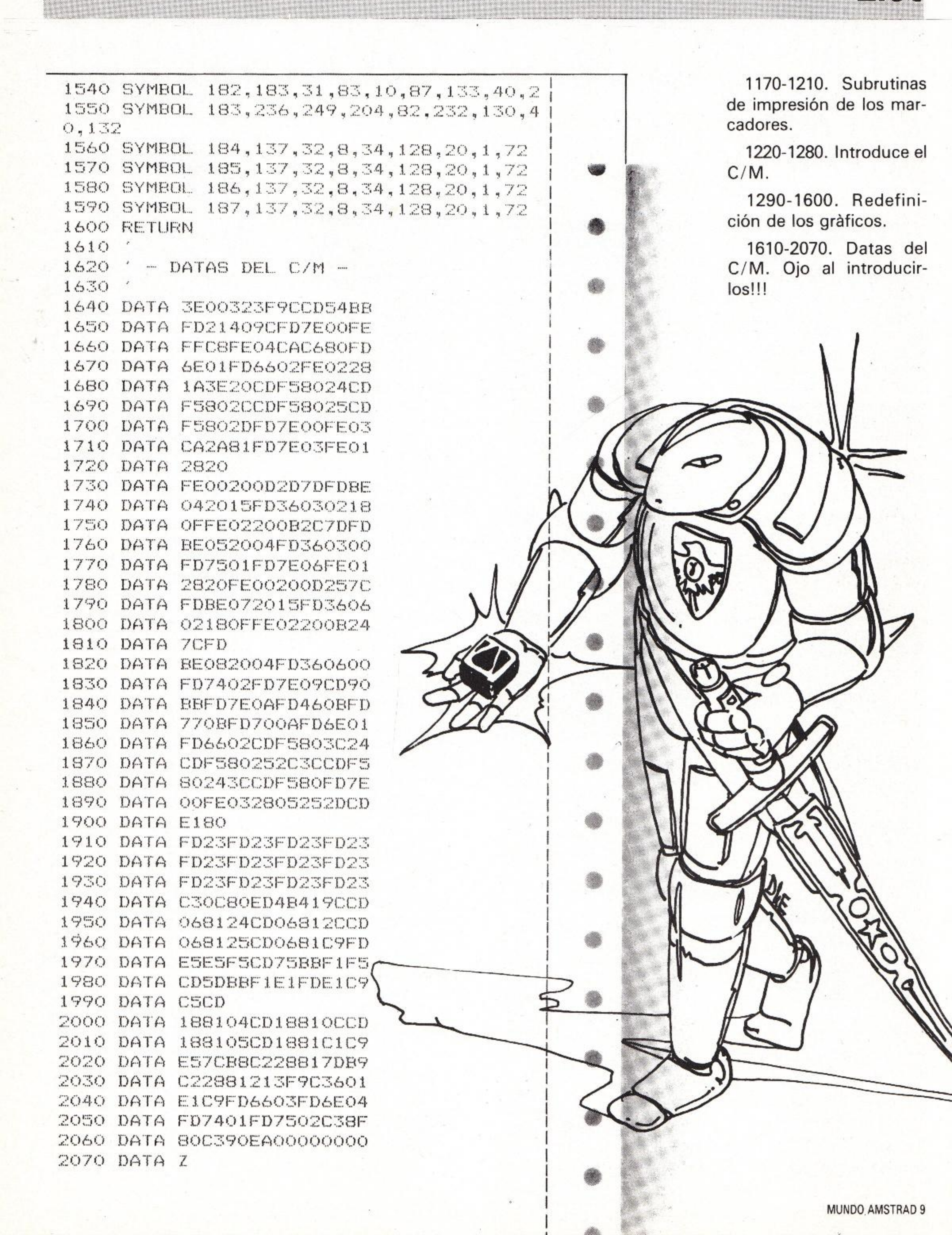
940-1030. Se pierde vida. Si son cero se imprime GAME OVER.

1040-1160. DATAS de las posiciones iniciales de todos los gráficos.

```
1220 ' - INTRODUCE C/M -
1230 '
1240 RESTORE 1640:D=32768
1250 READ A$:N=LEN(A$):IF A$="Z" THEN RE
TURN
1260 FOR A=1 TO N STEP 2:B$="&"+MID$(A$,
A,2)
1270 POKE D,VAL(B$):D=D+1:NEXT A:GOTO 12
50
1280 '
1290 ' - REDEFINICION DE GRAFICOS -
1300 '
1310 SYMBOL AFTER 160
1320 SYMBOL 160,127,195,131,159,155,255,
127,48
1330 SYMBOL 161,254,195,193,249,217,255,
254,12
1340 SYMBOL 162,124,255,207,10,15,15,30,
62
1350 SYMBOL 163,62,254,247,83,243,248,12
4,0
1360 SYMBOL 164,127,255,131,159,155,255,
127,48
1370 SYMBOL 165,254,255,193,249,217,255,
254,12
1380 SYMBOL 166,124,127,239,202,207,31,6
2,0
1390 SYMBOL 167,62,255,243,80,240,240,12
0,124
1400 SYMBOL 168,255,255,192,192,192,199,
199,192
1410 SYMBOL 169,255,255,3,3,3,227,255,31

1420 SYMBOL 170,192,199,199,192,192,192,
255,255
1430 SYMBOL 171,31,255,227,3,3,3,255,255

1440 SYMBOL 172,0,15,49,103,79,223,159,1
91
1450 SYMBOL 173,0,240,252,254,254,255,25
5,255
1460 SYMBOL 174,255,255,255,127,127,63,1
5,0
1470 SYMBOL 175,255,255,255,254,254,252,
240,0
1480 SYMBOL 176,7,28,49,35,103,79,95,95
1490 SYMBOL 177,224,248,252,252,254,254,
254,254
1500 SYMBOL 178,127,127,127,127,63,63,31
,7
1510 SYMBOL 179,254,254,254,254,252,252,
248,224
1520 SYMBOL 180,33,20,65,23,74,51,159,55
1530 SYMBOL 181,64,20,161,234,80,202,248
,237
```



```
1540 SYMBOL 182,183,31,83,10,87,133,40,2
1550 SYMBOL 183,236,249,204,82.232,130,4
0,132
1560 SYMBOL 184,137,32,8,34,128,20,1,72
1570 SYMBOL 185,137,32,8,34,128,20,1,72
1580 SYMBOL 186,137,32,8,34,128,20,1,72
1590 SYMBOL 187,137,32,8,34,128,20,1,72
1600 RETURN
1610 '
1620 ' - DATAS DEL C/M -
1630 '
1640 DATA 3E00323F9CCD54BB
1650 DATA FD21409CFD7E00FE
1660 DATA FFC8FE04CAC680FD
1670 DATA 6E01FD6602FE0228
1680 DATA 1A3E20CDF58024CD
1690 DATA F5802CCDF58025CD
1700 DATA F5802DFD7E00FE03
1710 DATA CA2A81FD7E03FE01
1720 DATA 2820
1730 DATA FE00200D2D7DFDBE
1740 DATA 042015FD36030218
1750 DATA 0FFE02200B2C7DFD
1760 DATA BE052004FD360300
1770 DATA FD7501FD7E06FE01
1780 DATA 2820FE00200D257C
1790 DATA FDBE072015FD3606
1800 DATA 02180FFE02200B24
1810 DATA 7CFD
1820 DATA BE082004FD360600
1830 DATA FD7402FD7E09CD90
1840 DATA BBFD7E0AFD460BFD
1850 DATA 770BFD700AFD6E01
1860 DATA FD6602CDF5803C24
1870 DATA CDF580252C3CCDF5
1880 DATA 80243CCDF580FD7E
1890 DATA 00FE032805252DCD
1900 DATA E180
1910 DATA FD23FD23FD23FD23
1920 DATA FD23FD23FD23FD23
1930 DATA FD23FD23FD23FD23
1940 DATA C30C80ED4B419CCD
1950 DATA 068124CD06812CCD
1960 DATA 068125CD0681C9FD
1970 DATA E5E5F5CD75BBF1F5
1980 DATA CD5DBBF1E1FDE1C9
1990 DATA C5CD
2000 DATA 188104CD18810CCD
2010 DATA 188105CD1881C1C9
2020 DATA E57CB8C228817DB9
2030 DATA C22881213F9C3601
2040 DATA E1C9FD6603FD6E04
2050 DATA FD7401FD7502C38F
2060 DATA 80C390EA00000000
2070 DATA Z
```

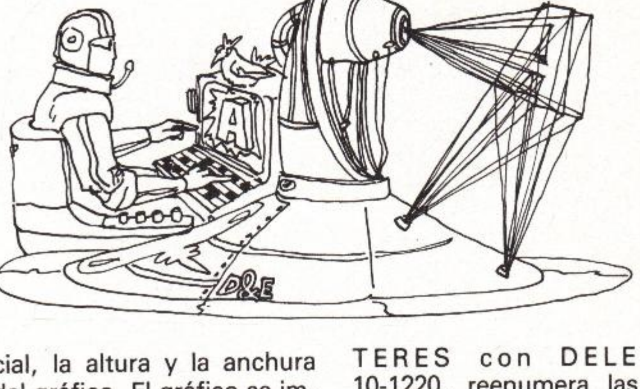
1170-1210. Subrutinas de impresión de los marcadores.

1220-1280. Introduce el C/M.

1290-1600. Redefinición de los gràficos.

1610-2070. Datas del C/M. Ojo al introducirlos!!!

GENERADOR DE CARACTERES GRAFICOS



Lo prometido es deuda, y como te indicamos el mes pasado, aquí te ofrecemos el programa mediante el cual podréis modificar el juego de caracteres gráficos de tu AMSTRAD.

El programa en sí es muy sencillo como podrás comprobar por su longitud, pero incorpora numerosas opciones que facilitan la edición de gráficos. Las posibilidades accesibles desde el teclado son:

CURSORES. Las flechas del cursor sirven para desplazar el "cursor gráfico" a través de la rejilla de diseño.

ENTER. Activa el punto de la celdilla, sobre la que se halla el cursor gráfico. Es decir, lo pinta de tinta (el bit correspondiente es punto a 1 = on).

DEL. Borra el punto de la rejilla posicionado por el cursor. Lo pinta de color del papel poniendo el bit correspondiente a 0 (posición off).

S. Graba con el nombre del fichero que le proporcionemos, el banco gráfico que actualmente reside en la memoria.

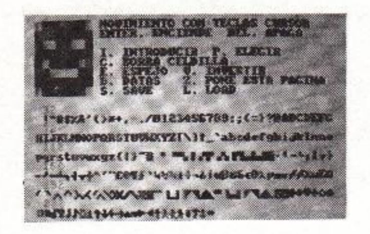
L. Carga el banco gráfico con el nombre que le demos.

C. Limpia totalmente la parrilla de diseño (CLS), poniendo todos los datos a cero. Los puntos toman el color del papel.

I. Introduce el gráfico que hemos definido en la celdilla en el carácter cuyo código indiquemos. Los datos en binario son transformados a decimal, introducción SYMBOL.

P. Traspasa el gráfico de la memoria a la celdilla, permitiéndonos modificarlo. Los datos son sacados de la memoria e interpretados como puntos de gráficos.

E. Hace un "espejo" del gráfico en pantalla. Por ejemplo, si tenemos una flecha que apunta en dirección izquierda (::) y seleccionamos esta opción, se transformará en una flecha que



mirará en sentido opuesto a la anterior.

Z. Suprime una ventana de instrucciones, donde se recuerdan brevemente las teclas y las funciones a las que corresponden.

V. Con esta opción obtenemos en la rejilla de diseño el inverso del gráfico que teníamos. Lo que se hace es borrar los puntos que están activados pintar los puntos que tienen el color del papel.

X. Nos permite presentar en pantalla un gráfico cuyas dimensiones excedan de 1 carácter. Nosotros le proporcionamos el carácter inicial, la altura y la anchura del gráfico. El gráfico se imprime línea a línea y de izquierda a derecha.

D. Esta es la opción más potente del programa, imprime en pantalla los SYMBOLS necesarios para definir los gráficos comprendidos en el margen que la introducimos. Este margen no puede ser superior a 22 líneas debido a que la pantalla, en caso contrario hará scroll y perderemos la posibilidad de copiar las líneas que se pierdan por la parte superior de la pantalla.

Una vez las líneas estén impresas en la pantalla el programa se detiene. Hemos de tener en cuenta, que las líneas no se introducen en la memoria. La forma de meterlas en la memoria en forma de programa BASIC, consiste en mover el cursor hasta el comienzo de cada línea, pulsar la tecla COPY para irla duplicando (no hace falta editarla en otro sitio, podemos hacerla sobre la que ya existe en pantalla). Una vez completa se pulsa ENTER para que pase a formar parte del programa BASIC.

Las líneas del programa toman números de línea a partir de 4 a 9000. Cuando quieras grabar este fragmento de programa, borra las líneas que constituyen el GENERADOR DE CARACTERES con DELETE 10-1220, reenumera las líneas para situarlas en la zona que quieras y grábalo. Para unirlo posteriormente puedes empezar a teclear tu programa teniendo introducido éste o hacer un MERGE con el bloque grabado.

O. Produce un efecto de animación en MODO 1, mostrando varios "gráficos" consecutivamente hasta la pulsación de la tecla 9. Consideramos que un gráfico es un conjunto de caracteres de n líneas por n columnas.

8. Efecto de animación en MODO 0. La pantalla es borrada durante la animación y restaurada al pulsar la tecla 9 de retorno.

7. Efecto de animación en MODO 2.

Comentario del programa

El programa consta de una única pantalla en la cual aparecen:

— La celdilla de edición.

— La ventana de introducciones (representación de opciones de animación) y entrada de datos, y;

— Exhibición de los caracteres para conocer cuales están definidos y cuales no, encabezados por un nú-

mero que es el código del primer carácter de la línea. Tanto para sacar, introducir o animar caracteres nos será necesario proporcionar su código.

Pero pasemos al comentario estructural del programa:

10-90. Se fija el comienzo del bloque de gráficos en la dirección 25000 para sernos posible leer y modificar los valores. La dirección de comienzo se guarda en la variable GRF, y en GRF2 el mismo valor menos 256 para eliminar los primeros 32 caracteres.

100-140. Pantalla en MODO 1 (texto 40columnas - 4 colores) y se establecen los colores para las tintas. Además se colorea el borde y se definen 3 ventanas para desempeñar las funciones que antes comentábamos.

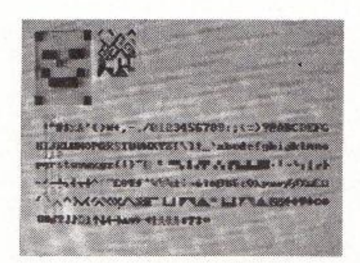
150-200. Se imprimen las instrucciones. Se dimensionan dos matrices, una de caracteres donde almacenar ceros y unos que luego se transformarán en números en decimal. Y una tabla que contiene los 8 bytes del gráfico. A continuación se dibuja la tabla de caracteres, se ponen a 0 todos los caracteres de la matriz A$ y se fijan las coordenadas X, y del cursor con sus valores iniciales.

210-480. Bucle principal del programa. Se sitúa el cursor en sus coordenadas y se leen todas las teclas mediante las cuales se accede a las opciones del programa, situadas en subrutinas llamadas con la instrucción GOSUB.

Aquí se incluyen algunas de las opciones cuya brevedad no pide el desarrollo de una subrutina externa al bucle principal. Estas son las opciones de GRABACION, CARGA y BORRADO DE LA CELDILLA.

Queremos hacer notar la velocidad con que se ejecuta el programa, debido a esto, se ha incluido un FOR-NEXT que nos sirve como pausa, de forma que aquellos que prefieran más velocidad en la edición, tendrán que reducir el valor del bucle y los que deseen una edición más lenta, habrán de incrementar el valor de éste número.

490-600. Subrutina que imprime las líneas que pasadas a programa nos generarán los gráficos. Lo primero que hacemos es tomar los datos de los caracteres que



han de salir. Luego se imprimen las líneas SYMBOL y se detiene el programa con END. Recuerda que al pulsar esta opción todos los caracteres redefinidos se inicializarán. De forma que si quieres transformar a programa un número mayor de 22 caracteres, tendrás que hacerlo en varias veces, grabando en CASSETTE o DISCO el banco gráfico antes de pulsar esta opción.

610-690. Subrutina de introducción de un carácter. Se calculan los datos a partir de la tabla binaria, y se introducen con la instrucción SYMBOL.

770-810. Dibuja en pantalla el contenido de la matriz. Esta rutina es llamada después de cada transformación realizada en la tabla para que estos cambios queden representados en pantalla.

820-860. Subrutina de impresión de un gráfico.

870-900. Subrutina de espejo.

910-930. Subrutina de inversión.

940-1030. Impresión de la tabla de caracteres con sus códigos ASCII.

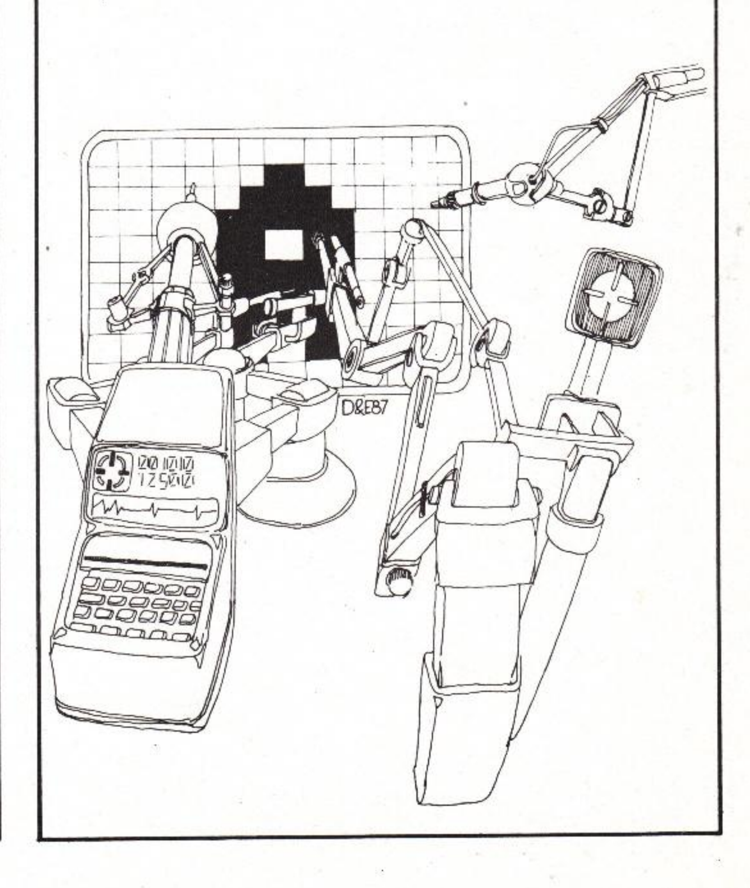
1040-1140. Impresión de las instrucciones.

1150-1220. Subrutina de animación.

```
10 ' *********************
20 ' *                   *
30 ' * GENERADOR GRAFICO *
40 ' *                   *
50 ' *********************
60 '
70 SYMBOL AFTER 256:MEMORY 24999:SYMBOL
AFTER 32
80 GRF=HIMEM+1
90 GRF2=GRF-256
100 MODE 1
110 INK 0,26:INK 1,0:INK 2,20:INK 3,8
120 WINDOW#1,1,8,1,8:WINDOW 1,40,9,25:WI
NDOW#2,10,40,1,8
130 PAPER#1,2:CLS#1:PEN#1,1
140 BORDER 26
150 GOSUB 1040
160 DIM A$(8,8):DIM G(8)
170 GOSUB 940
180 FOR A=1 TO 8:FOR B=1 TO 8:LET A$(A,B
)="0":NEXT B
190 NEXT A
200 X=1:Y=1
210 LOCATE#1,Y,X
220 IF A$(X,Y)="0" THEN PRINT#1," ";
230 IF A$(X,Y)="1" THEN PRINT#1,CHR$(143
);
240 IF INKEY(8)=0 THEN Y=Y-1:IF Y<1 THEN
 Y=1
250 IF INKEY(1)=0 THEN Y=Y+1:IF Y>8 THEN
 Y=8
260 IF INKEY(0)=0 THEN X=X-1:IF X<1 THEN
 X=1
```

```
270 IF INKEY(2)=0 THEN X=X+1:IF X>8 THEN
 X=8
280 IF INKEY(18)=0 THEN A$(X,Y)="1"
290 IF INKEY(79)=0 THEN A$(X,Y)="0"
300 LOCATE#1,Y,X
310 IF A$(X,Y)="1" THEN PRINT#1,CHR$(233
);
320 IF A$(X,Y)="0" THEN PRINT#1,CHR$(232
);
330 T$=INKEY$
340 IF INKEY(60)=0 THEN CLS #2:INPUT #2,
"NOMBRE:",N$:SAVE N$,B,GRF,224*8
350 IF INKEY(36)=0 THEN CLS #2:INPUT #2,
"NOMBRE:",N$:N$=N$+".BIN":LOAD N$,GRF:SO
UND 1,150
360 IF INKEY(61)=0 THEN GOSUB 490
370 IF INKEY(32)=0 THEN GOSUB 1150
380 IF INKEY(40)=0 THEN MODE 0:GOSUB 115
0:RUN 100
390 IF INKEY(41)=0 THEN MODE 2:GOSUB 115
0:RUN 100
400 IF INKEY(35)=0 THEN GOSUB 610
410 IF INKEY(62)=0 THEN CLS#1:GOTO 180
420 IF INKEY(27)=0 THEN GOSUB 700
430 IF INKEY(63)=0 THEN GOSUB 820
440 IF INKEY(58)=0 THEN GOSUB 870
450 IF INKEY(71)=0 THEN GOSUB 1040
460 IF INKEY(55)=0 THEN GOSUB 910
470 FOR Z=1 TO 60:NEXT Z
480 GOTO 210
490 CLS#2:PRINT#2,"*** DATAS ***":PRINT#
2:INPUT#2,"SYMBOL INICIAL:";SI:INPUT#2,"
SYMBOL FINAL:";SF
```

```
500 PEN 1:MODE 2
510 L=9000+(2*(SI-32))
520 SYMBOL AFTER 256
530 FOR N=SI TO SF
540 PRINT L+(5*(N-SI));"SYMBOL";N;CHR$(8
);",";
550 p=GRF2+(N*8)
560 FOR b=p TO P+7:z$=STR$(PEEK(b))
570 PRINT MID$(z$,2,LEN(z$)-1);",";
580 NEXT b:PRINT CHR$(8);" "
590 NEXT n
600 END
610 CLS#2:PRINT#2,"*** INTRODUCIR ***":P
RINT#2:INPUT#2,"EN QUE CARACTER (32-255)
:";C
620 PRINT#2,"CAMBIAMOS: ";CHR$(C)
630 FOR A=1 TO 8:D$="":FOR Z=1 TO 8:D$=D
$+A$(A,Z):NEXT Z:NU=VAL("&X"+D$)
640 G(A)=NU
650 POKE GRF2-1+(C*8)+A,NU:NEXT A
660 SYMBOL C,G(1),G(2),G(3),G(4),G(5),G(
6),G(7),G(8)
670 GOSUB 1010
680 PRINT#2,"OK. INTRODUCIDO";C;": ";CHR
$(C)
690 RETURN
700 CLS#2:PRINT#2,"*** ELEGIR ***":PRINT
#2:INPUT#2,"QUE CARACTER (32-255):";C
710 P=GRF2+(C*8):FOR Z=P TO P+7
720 Z$=BIN$(PEEK(Z),8):FOR N=1 TO 8:A$(Z
-P+1,N)=MID$(Z$,N,1):NEXT N
730 NEXT Z
740 GOSUB 770
750 PRINT#2,"OK, ";CHR$(C);" EN PANTALLA
."
760 RETURN
770 FOR K=1 TO 8:FOR L=1 TO 8
780 LOCATE#1,L,K
790 IF A$(K,L)="0" THEN PRINT#1,CHR$(32)
;
800 IF A$(K,L)="1" THEN PRINT#1,CHR$(143
);
810 NEXT L:NEXT K:RETURN
820 CLS #2:INPUT#2,"CARACTER INICIAL:";C
830 INPUT #2,"ANCHURA:",AN:INPUT #2,"ALT
URA:",AL
840 CLS#2
850 FOR XX=1 TO AL:FOR YY=1 TO AN:LOCATE
 #2,YY,XX:PRINT #2,CHR$(C):C=C+1:NEXT YY
:NEXT XX
860 RETURN
870 FOR K=1 TO 8:FOR L=1 TO 4
880 Z$=A$(K,L):A$(K,L)=A$(K,9-L):A$(K,9-
L)=Z$
890 NEXT L:NEXT K
900 GOSUB 770:RETURN
910 FOR K=1 TO 8:FOR L=1 TO 8
920 IF A$(K,L)="0" THEN A$(K,L)="1" ELSE
 A$(K,L)="0"
930 NEXT L:NEXT K:GOSUB 770:RETURN
940 H$="":FOR A=32 TO 71:H$=H$+CHR$(A):N
EXT A
950 I$="":FOR A=72 TO 111:I$=I$+CHR$(A):
NEXT A
960 J$="":FOR A=112 TO 151:J$=J$+CHR$(A)
:NEXT A
970 K$="":FOR A=152 TO 191:K$=K$+CHR$(A)
:NEXT A
980 L$="":FOR A=192 TO 231:L$=L$+CHR$(A)
:NEXT A
990 M$="":FOR A=232 TO 255:M$=M$+CHR$(A)
:NEXT A
1000 CLS:PRINT:FOR A=32 TO 255 STEP 40:P
EN 2:PRINT A:PRINT:NEXT A
1010 PEN 1:LOCATE 1,3:PRINT H$:PRINT I$:
PRINT J$:PRINT K$:PRINT L$:PRINT M$
1020 PEN 1
1030 RETURN
1040 CLS#2
1050 PRINT#2,"MOVIMIENTO CON TECLAS CURS
OR"
1060 PRINT#2,"ENTER. ENCIENDE  DEL. APAG
A"
1070 PRINT#2
1080 PRINT#2,"I. INTRODUCIR  P. ELEGIR"
1090 PRINT#2,"C. BORRA CELDILLA"
1100 PRINT#2,"E. ESPEJO  V. INVERTIR"
1110 PRINT#2,"D. DATAS   Z. PONE ESTA PA
GINA"
1120 PRINT#2,"S. SAVE    L. LOAD";
1130 SOUND 1,200,10
1140 RETURN
1150 CLS #2:INPUT #2,"CARACTER INICIAL:"
,CI:INPUT #2,"CUANTOS:",C
1160 INPUT #2,"ANCHURA:",AN:INPUT #2,"AL
TURA:",AL
1170 CLS #2
1180 CC=CI:FOR J=1 TO C
1190 FOR XX=1 TO AL:FOR YY=1 TO AN:LOCAT
E #2,YY,XX:PRINT#2,CHR$(CC):CC=CC+1:NEXT
 YY:NEXT XX:FOR O=1 TO 100:NEXT O:NEXT J
1200 IF INKEY(33)=0 THEN RETURN
1210 IF INKEY(49)=0 THEN GOTO 1150
1220 GOTO 1180
```



ORGANO ELECTRONICO

```
10  '>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >>>>> A.G.Chico F.dez de Teran >>>>>
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
20  '-------------inicializacion----------
30 KEY 0,"speed key 20,1"
40 DIM p(25)
50 FOR n=1 TO 20
60 READ periodo
70 p(n)=periodo
80 NEXT
90 t=30
100 DATA 159,150,142,134,127,119,113,106
,100,95,89,84,80,75,71,67,63,60,56,53
110 CLS:INPUT ;"con acomp.(s/n)?";a$:CLS
:GOSUB 5040:SPEED KEY 1,1:IF a$="s"THEN
GOTO 430 ELSE GOTO 5010
120 GOTO 110
130  '--------- lectura del teclado -------
140 i$=INKEY$
150 v=0
160 IF i$="a" THEN s=1:v=7
170 IF i$="w" THEN s=2:v=7
180 IF i$="s" THEN s=3:v=7
190 IF i$="e" THEN s=4:v=7
200 IF i$="d" THEN s=5:v=7
210 IF i$="f" THEN s=6:v=7
220 IF i$="t" THEN s=7:v=7
230 IF i$="g" THEN s=8:v=7
240 IF i$="y" THEN s=9:v=7
250 IF i$="h" THEN s=10:v=7
260 IF i$="j" THEN s=11:v=7
270 IF i$="i" THEN s=12:v=7
280 IF i$="k" THEN s=13:v=7
290 IF i$="o" THEN s=14:v=7
300 IF i$="l" THEN s=15:v=7
310 IF i$="p" THEN s=16:v=7
320 IF i$=":" THEN s=17:v=7
330 IF i$=";" THEN s=18:v=7
340 IF i$="[" THEN s=19:v=7
350 IF i$="]" THEN s=20:v=7
360 IF i$="" THEN i$=" "
370 IF ASC(i$)>48 AND ASC(i$)<54 THEN z$
=i$:LOCATE 22,13:PRINT z$
380 IF NOT INKEY(1) AND t>2 THEN t=t-2:L
OCATE 15,16:PRINT"TEMPO:";60-T
390 IF NOT INKEY(8) AND t<58 THEN t=t+2:
LOCATE 15,16:PRINT"TEMPO:";60-T
400  '-------------sonido-----------------
410 SOUND 1,p(s),3,v
420 RETURN
430  '----------------acomp----------------
440 g=1:z$="1"
450 LOCATE 15,13:PRINT"ACOMP.:";z$
460 LOCATE 15,16:PRINT"TEMPO:";60-T
470 EVERY 4,1 GOSUB 140
480 g=-(1*(z$="1")+2*(z$="2")+3*(z$="3")
```

Con este programa convertirás tu AMSTRAD en un órgano electrónico. Podrás interpretar melodías directamente desde el teclado con o sin acompañamiento. Hay cinco acompañamientos distintos, que podrás seleccionar pulsando las teclas con los números correspondientes (1, 2, 3, 4, 5 ó f1, f2, f3, f4, f5).

La velocidad del acompañamiento (tempo) se puede seleccionar también con los cursores. Asimismo, la velocidad de autorrepetición de las teclas está alterada. Para poder escribir al salir del programa, f0 tiene definida la velocidad normal, con lo que pulsando esta tecla más ENTER todo volverá a la normalidad.

DESCRIPCION DEL PROGRAMA

Inicialización: (20-120). Se define la velocidad de autorrepetición del teclado. Se introducen los valores de los periodos de las notas en una matriz. Se pregunta si se quiere acompañamiento o no y se salta a la rutina correspondiente, pasando antes por la subrutina de pantalla.

Rutina de acompañamiento: (450-490). Se establecen los valores iniciales del acompañamiento. Se pone en marcha la subrutina de lectura del teclado por interruptores y melodía, y se salta al acompañamiento que corresponda según los datos suministrados por dicha rutina: la variable g contiene el número de acompañamiento, (*) y t la duración. Inicialmente, su valor es de 30. Este valor se toma como base en una escala de 0 a 60, escala que se obtiene restando a 60 el valor t. De estar forma, con el valor de 60, el tempo alcanza su valor más rápido, y viceversa para el 0.

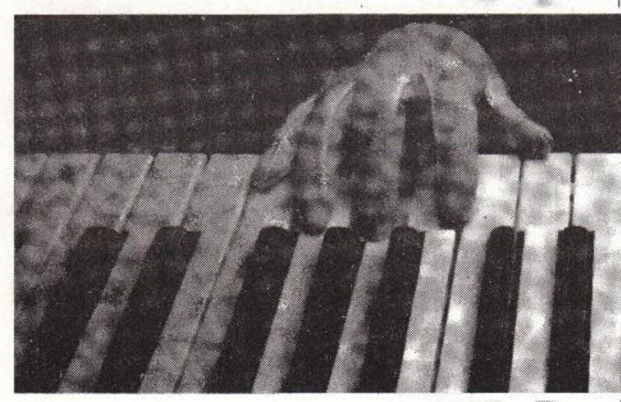
Lectura del teclado: (130-420). Esta rutina es llamada mediante interrupciones por la rutina de acompañamiento. Se divide en la de lectura del teclado en sí y la de melodía. La primera consiste en una *lista* de sentencias if que comprueban las teclas del órgano una por una y le asigna a la variable S un valor que se corresponde con el valor del periodo de la nota en la matriz p (). Nótese que todas las sentencias if se podían haber sustituido por una sola sentencia del tipo:

Organo electrónico

s = 1*(i$ = "A") + Z*
(i$ = "W") + ...

Sin embargo, estas sentencias son más rápidas de teclear mediante el AUTO y el COPY, ya que lo único que varía de una sentencia a otra es la letra de la tecla y el valor S.

La sentencia "if" correspondiente a la lectura del número del acompañamiento se asegura de que i$ es una de las teclas del 1-5, antes de cambiar el valor a g (que era la variable que indicaba el número del acompañamiento). Esto se hace con objeto de que con sólo pulsar una de las cinco teclas, el acompañamiento cambie (de otra forma, al ir tocando la melodía, para cambiar de acompañamiento tendríamos que mantener la tecla de acompañamiento correspondiente pulsada hasta que terminase la comprobación de todas las teclas, sino g quedaría con el último valor chequeado). La sentencia 360 evita que al intentar averiguar el código ascII de la cadena"" (vacío) cuando no se esté pulsando ninguna tecla, en la 370, el ordenador nos de mensaje de error. Como se ve, el



método seguido para que no suenen las notas cuando no se estén pulsando las teclas es el de poner el volumen a 0. Así, la nota se seguirá tocando pero no podremos oirla.

La otra subrutina, SONIDO (410-420), lo único que hace es tocar la nota con los datos de periodo (s) y volumen (v).

Se le da una duración muy pequeña para que las notas no se almacenen en la cola de sonido, retardando así la respuesta al teclado.

(500-1380).— Aquí están las cinco subrutinas de los cinco

```
+4*(z$="4")+5*(z$="5"))
490 ON g GOTO 500,690,870,1050,1220
500 '--------do mayor-----------------
510 SOUND 6,239,t,7
520 FOR n=1 TO 5*t:NEXT
530 SOUND 2,190,t/2,7
540 SOUND 4,159,t/2,7
550 FOR n=1 TO 5*t:NEXT
560 SOUND 2,190,t/2,7
570 SOUND 4,159,t/2,7
580 FOR n=1 TO 5*t:NEXT
590 SOUND 6,319,t,7
600 FOR n=1 TO 5*t:NEXT
610 SOUND 2,190,t/2,7
620 SOUND 4,159,t/2,7
630 FOR n=1 TO 5*t:NEXT
640 SOUND 2,190,t/2,7
650 SOUND 4,159,t/2,7
660 FOR n=1 TO 5*t:NEXT
670 GOTO 480
680 '------------ re menor -----------
690 SOUND 6,213,t,7
700 FOR n=1 TO 5*t:NEXT
710 SOUND 2,179,t/2,7
720 SOUND 4,142,t/2,7
730 FOR n=1 TO 5*t:NEXT
740 SOUND 2,179,t/2,7
750 SOUND 4,142,t/2,7
760 FOR n=1 TO 5*t:NEXT
770 SOUND 6,284,t,7
780 FOR n=1 TO 5*t:NEXT
790 SOUND 2,179,t/2,7
800 SOUND 4,142,t/2,7
810 FOR n=1 TO 5*t:NEXT
820 SOUND 2,179,t/2,7
830 SOUND 4,142,t/2,7
840 FOR n=1 TO 5*t:NEXT
850 GOTO 480
860 '---------------sol mayor----------
870 SOUND 6,319,t,7
880 FOR n=1 TO 5*t:NEXT
890 SOUND 2,253,t/2,7
900 SOUND 4,213,t/2,7
910 FOR n=1 TO 5*t:NEXT
920 SOUND 2,253,t/2,7
930 SOUND 4,213,t/2,7
940 FOR n=1 TO 5*t:NEXT
950 SOUND 6,426,t,7
960 FOR n=1 TO 5*t:NEXT
970 SOUND 2,253,t/2,7
980 SOUND 4,213,t/2,7
990 FOR n=1 TO 5*t:NEXT
1000 SOUND 2,253,t/2,7
1010 SOUND 4,213,t/2,7
1020 FOR n=1 TO 5*t:NEXT
```

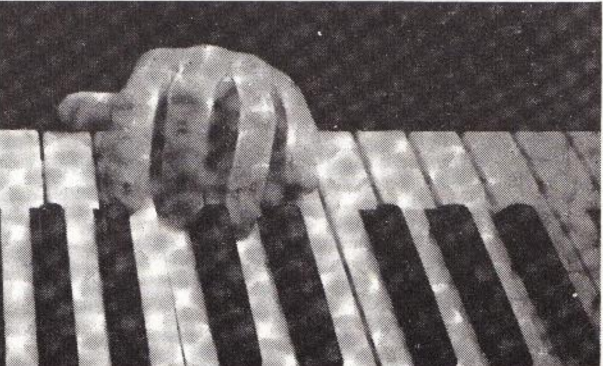
```
1030 GOTO 480
1040 '---------- la menor ----------------
1050 SOUND 6,284,t,7
1060 FOR n=1 TO 5*t:NEXT
1070 SOUND 2,239,t/2,7
1080 SOUND 4,190,t/2,7
1090 FOR n=1 TO 5*t:NEXT
1100 SOUND 2,239,t/2,7
1110 SOUND 4,190,t/2,7
1120 FOR n=1 TO 5*t:NEXT
1130 SOUND 6,379,t,7
1140 FOR n=1 TO 5*t:NEXT
1150 SOUND 2,239,t/2,7
1160 SOUND 4,190,t/2,7
1170 FOR n=1 TO 5*t:NEXT
1180 SOUND 2,239,t/2,7
1190 SOUND 4,190,t/2,7
1200 FOR n=1 TO 5*t:NEXT
1210 GOTO 480
1220 '---------fa mayor-----------------
1230 SOUND 6,358,t,7
1235 FOR n=1 TO 5*t:NEXT
1240 SOUND 2,284,t/2,7
1250 SOUND 4,239,t/2,7
1260 FOR n=1 TO 5*t:NEXT
1270 SOUND 2,284,t/2,7
1280 SOUND 4,239,t/2,7
1290 FOR n=1 TO 5*t:NEXT
1300 SOUND 6,478,t,7
1310 FOR n=1 TO 5*t:NEXT
1320 SOUND 2,284,t/2,7
1330 SOUND 4,239,t/2,7
1340 FOR n=1 TO 5*t:NEXT
1350 SOUND 2,284,t/2,7
1360 SOUND 4,239,t/2,7
1370 FOR n=1 TO 5*t:NEXT
1380 GOTO 480
5000 '---------- sin acom.--------------
5010 EVERY 4,1 GOSUB 140
5020 GOTO 5020
5030 '----------- pantalla ------------
5040 MODE 1
5050 PAPER #2,2:WINDOW #2,7,31,5,9:CLS#2
5060 LOCATE 9,6:PRINT"w e   t y  i o p
   ["
5070 LOCATE 8,7:PRINT"
     "
5080 LOCATE 8,8:PRINT"a s d f g h j k l
 : ; ]"
5090 LOCATE 15,4:PEN 3:PRINT"TECLADO"
5100 LOCATE 14,20:PRINT CHR$(164);CHR$(6
5);CHR$(46);CHR$(71);CHR$(46);CHR$(32);C
HR$(67);CHR$(72);CHR$(73);CHR$(67);CHR$(
79)
5110 RETURN
```

acompañamientos distintos. Tienen ritmo de valls y están pensados para acompañar a una melodía que esté en DOM (ya que ésta tonalidad, al no tener semitonos, es la más fácil de interpretar).

Combinando adecuadamente en cada subrutina de acompañamiento las diferentes notas que las componen, se podrán construir ritmos diferentes aumentando así el número de acompañamientos (por ejemplo: como se observará, cada rutina de acompañamiento se compone de una sound, un bucle de pausa, y dos pares de SOUND con su correspondiente bucle de espera. Si se elimina uno de estos pares con su bucle, se obtendrá un ritmo de marcha).

Asimismo, con unos mínimos conocimientos de música se podría variar la tonalidad del acompañamiento cambiando únicamente las notas (periodos) de las sentencias sound.

De esta forma, utilizando teclas numéricas que no se usan y la fila de debajo del teclado de la melodía (de la z a la], podéis ampliar el número de acompañamientos hasta 23.



Sin acompañamiento: (5000-5020). Activa las interrupciones y se queda en un bucle sin fin.

Pantalla: Define una pantalla para representar un teclado de piano con las letras del teclado. Así, la letra A corresponde a la nota SOL, la tecla W a SOL♯, etc. Obsérvese que en la fila a -] están las notas naturales, y en la W [las sostenidas o bemoles.

Los que tengáis un ordenador AMSTRAD 6128 con teclado en castellano, tenéis que cambiar los ":" de éste listado por la tecla correspondiente.

TRANSMUTER

Lejos, en el futuro el sol ha envejecido y se ha convertido en una GIGANTE ROJA. El hombre ha tenido que cobijarse bajo tierra, utilizando el calor del planeta para sobrevivir.

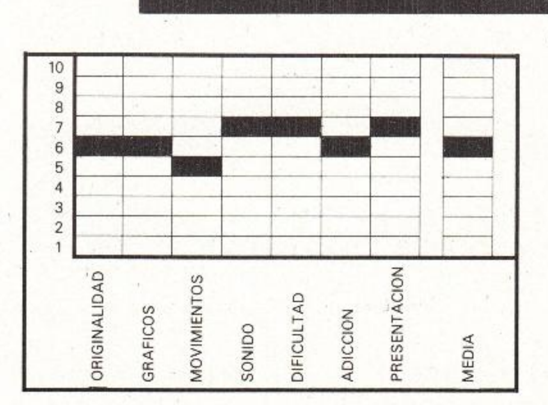
Generación tras generación se ha ido cobijando cada vez a mayor profundidad dejando en cada nivel anterior mortíferas máquinas cubriendo la retaguardia. Ultimamente ha sido creada una nave para obtener el abundante mineral de la corteza terrestre y los últimos descendientes humanos pueden abandonar el moribundo planeta. Unicamente las máquinas luchadoras permanecen todavía guardando a los humanos que hace tantos años que las colocaron.

Tú eres el capitán TENZ, uno de los primeros hombres nacidos en la primera colonia establecida fuera de la tierra. Tu misión es regresar al tu planeta donde tus antepasados instalaron sus trampas mortíferas, para rescatar a algún viajero espacial.

TRANSMUTER es un juego de acción con numerosos escenarios que hacen que la dificultad sea cada vez mayor a medida que avanzamos en nuestra aventura.

La presentación cuenta con un atractivo dibujo de portada y una melodía muy bien realizada.

Los gráficos son abundantes y variados, sin embargo el SCROLL (9 continuo hacia la izquierda) es muy brusco y hace que nuestra nave se mueva de un modo parpadeante por la pantalla. Algunos escenarios poseen una buena calidad gráfica, en cambio otros resultan un poco pobres. La idea del juego en general es buena y las situaciones se desarrollan con gran velocidad lo que convierte a TRANSMUTER en un juego divertido y adictivo.



DRUID

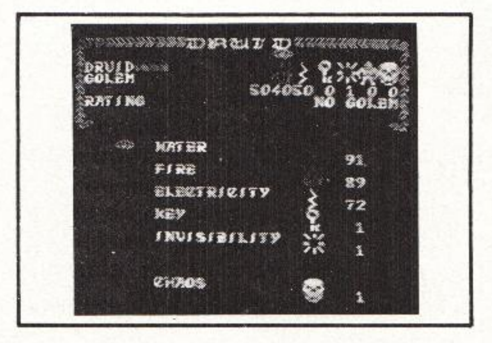
DRO-SOFT

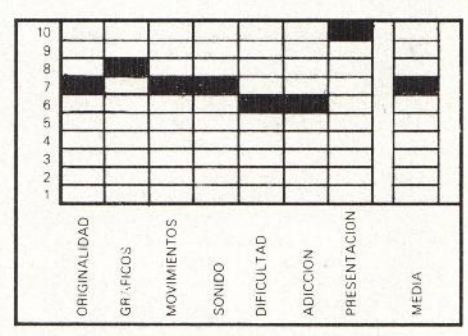
Hasta ahora se ha mantenido el equilibrio político y ha habido paz por todo Berlon. Pero a través de una puerta inter-dimensional en las mazmorras del maléfico nigromante Acamantor, han aparecido ahora cuatro principes endemoniados.
Ha recaido sobre ti, el último de los Grandes Druidas, la tarea de destruir a los principes y de cerrar la puerta.
Los principes endemoniados andan acechando por las tenebrosas profundidades de la mazmorra, y solamente pueden destruirse por medio de tu poderoso hechizo, que en este caso debe de transportarse por medio del tacto.

Por toda la mazmorra encontrarás arcas que contienen hechizos de una fuerza formidable para ayudarte en tu búsqueda y Pentogramas de vida, que revitalizarán totalmente tu esencia de vida. Usa estos recursos con prudencia a fin de sobrevivir el ataque constante de los monstruos dimensionales infernales y poder hacer frente a los principes endemoniados.

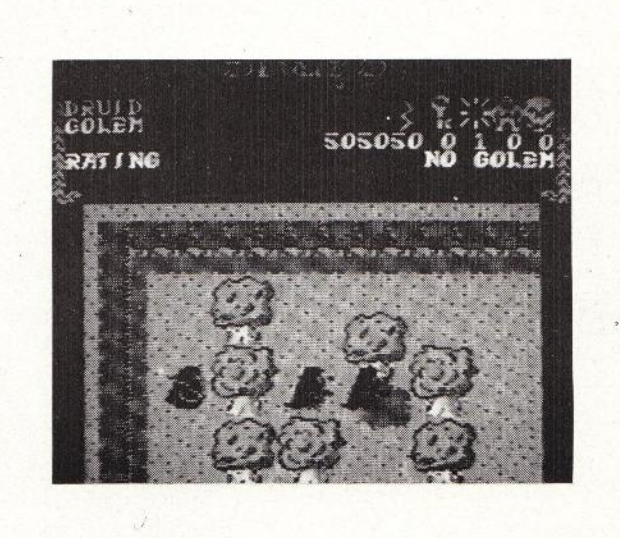
Triunfa y alcanzarás el alto puesto de Maestro de la luz, el mayor de todos los Druidas. Fracasa y no serás más que un mediocre ingenio y tu alma cautiva se reunirá con los numerosos fantasmas de los que te han precedido, perdidas en las mazmorras de Acamantor.

**DRUID es un juego en el que tendrás que recorrer un intrincado laberinto buscando los hechizos que te ayuden a destruir a las fuerzas del mal.
La presentación es muy espectacular, ya que junto a un buen dibujo de portada, al mismo tiempo que al efectuar la carga del programa, podemos escuchar una melodía muy bien adaptada para el ordenador.
DRUID tiene una gran abundancia de gráficos muy elaborados, sin embargo el espacio "útil" donde se desarrolla el juego es algo reducido. En él, el laberinto se desplaza mediante un "SCROLL" en cuatro sentidos muy suave.
Los sonidos de ambiente, los movimientos y la rapidez de respuesta son otra buena característica de DRUID.
La dificultad no es muy elevada pero hay que estar muy atento a los múltiples enemigos que van apareciendo en pantalla a lo largo de la aventura.**

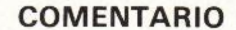




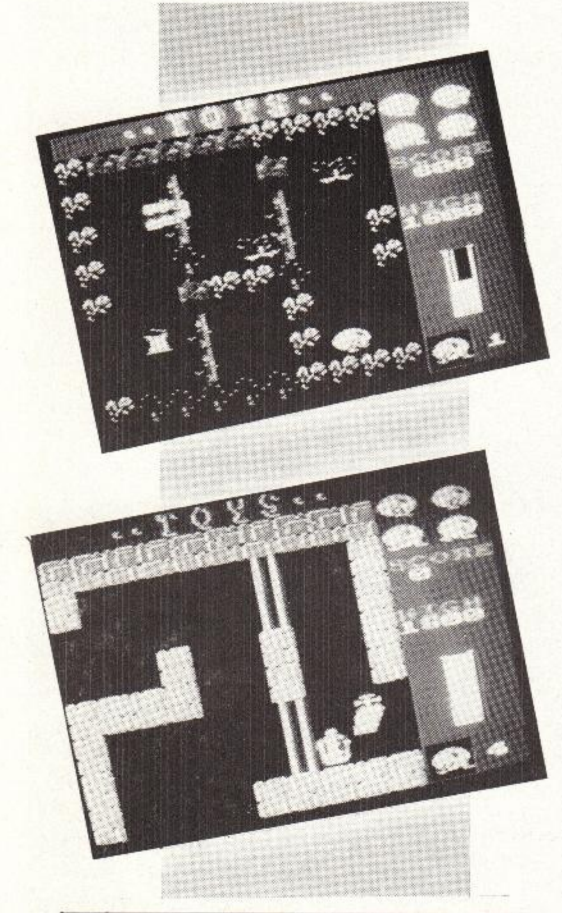




TOYS



COMENTARIO



El tema de TOYS trata de una rebelión de juguetes contra CIBER dueño, por no hacerles caso y preferir jugar con ordenadores.
La aventura transcurre en tres fases o laberintos diferentes (un jardín, un castillo y un cementerio) de 15 pantallas cada uno, donde los juguetes han desperdigado el material informático de CIBER.
Los gráficos de TOYS están bien elaborados siendo todos los juguetes animados y de varios colores, sin embargo la diversidad de estos no es muy grande.
Los movimientos son rápidos y siempre existen varios objetos en movimiento dentro de cada pantalla: la respuesta de CIBER AL TECLADO es también muy rápida.
La pantalla de presentación no es demasiado atractiva y cuenta con una corta melodía inicial: el resto de sonidos del juego cumplen bien su misión de ambientar.
El nivel de dificultad es elevado y esto hace que el juego resulte divertido aunque no muy emocionante.

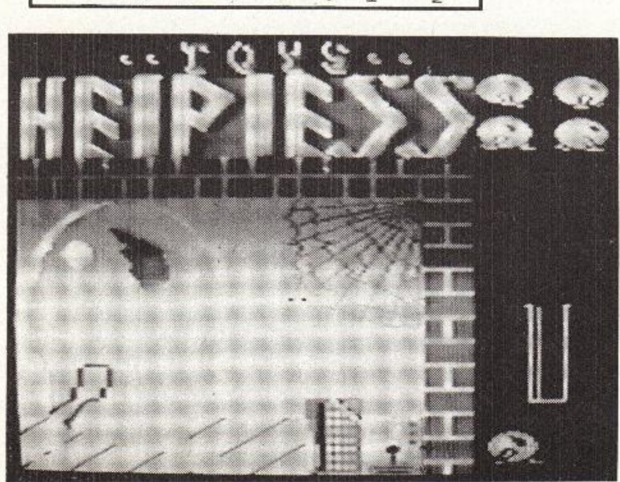
ORIGINALIDAD	6
GRAFICOS	7
MOVIMIENTOS	5
SONIDO	6
DIFICULTAD	7
ADICCION	7
PRESENTACION	6
MEDIA	6

Nuestro amigo CIBER es un gran entusiasta de los ordenadores y por culpa de su afición tiene abandonados al resto de sus juguetes.

Un buen día CIBER, como siempre, va a su cuarto a jugar y se lo encuentra vacio. Ni sus ordenadores, ni ninguno de sus juguetes se encuentra en la habitación.

Sorprendido CIBER baja a la calle y se encuentra con que sus juguetes, hartos de ser ignorados, se han sublevado contra él y han esparcido sus ordenadores y sus juegos por diversas partes del jardín, en las ruinas de un viejo castillo y en el cementerio cercano.

Debes ayudar a CIBER a rescatar sus ordenadores y todo su material informático secuestrado por los juguetes en plena rebelión. La tarea no será sencilla pues todos los juguetes han tomado vida y atacarán a CIBER allá donde lo encuentren, restándole energía hasta acabar con sus cuatro vidas.







THE SENTINEL

El objeto del juego es absorber al "Sentinel" y sustituirlo como regidor del panorama. Una vez logrado, el jugador puede supresspaciar a un nuevo mundo y tratar de lograrlo de nuevo.

Una vez que se haya insertado el cassette con el juego, aparecerá el título en la pantalla. Al pulsar cualquier tecla se le pedirá que inscriba un número de panorama, del 0000 a 9999.

Una vez inscrito el número de panorama y pulsar retorno, se le pedirá que marque un número de clave de 8 cifras, al menos que haya elegido panorama 0000, que no lo requiere.

Pasados unos segundos, aparecerá la superficie panorámica. El Sentinel y los Sentries permanecerán inactivos hasta que el jugador gaste o absorba energía. Esto le permitirá mirar alrededor y plantear su ataque al Sentinel.

Una vez activados, el Sentinel y los Sentries giran despacio, barriendo el panorama, buscando cuadros que contienen más de 1 unidad de energía. Si pueden ver con claridad uno de estos cuadros, el Sentinel/Sentry reducirá la engergiá a 1 unidad, absorbiendo 1 unidad a la vez y creando un árbol al azar en el panorama. De esta forma, el robot se convierte en bloque y el bloque se convierte en árbol.

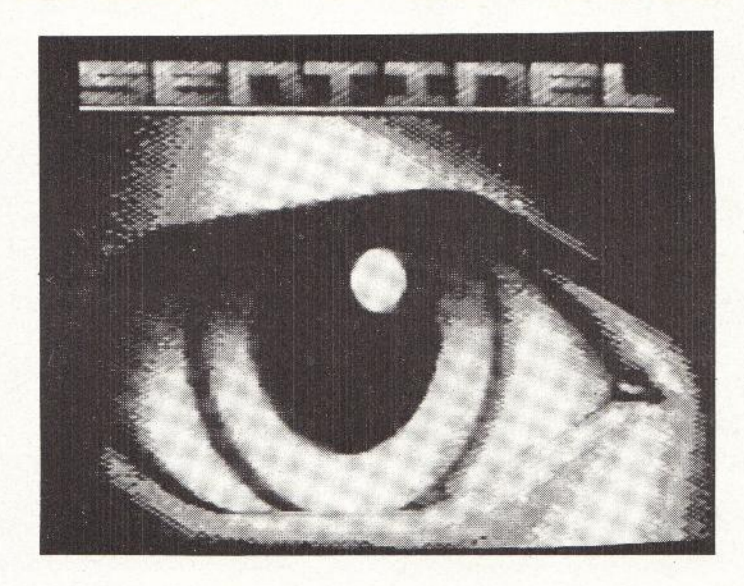
Se absorbe y se crean cosas, haciendo girar las miras y centrándolas en la superficie del cuadro bajo el objeto a ser absorbido/creado. Sin embargo los bloques actúan como extensión de la superficie del cuadro, y las miras deberan ser dirigidas al costado del bloque, se pueden apilar los bloques y colocarse cosas sobre ellos.

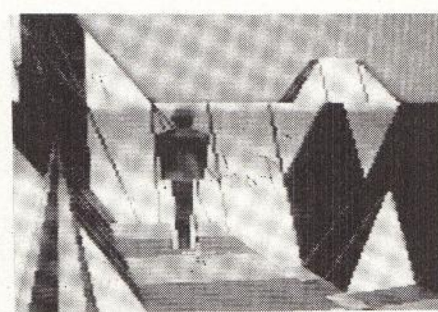
La energía disponible aparece mostrada en la parte superior izquierda de la pantalla en forma de iconos de robot, bloque y árbol. Un robot de color (no azul) vale por 15 unidades de energía.

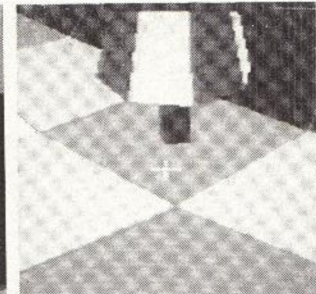
Para moverse alrededor del panorama, se debe crear un robot y entonces, con las mairas aún encendidas, púlsese la tecla de transferencia. Así se pasará al nuevo robot, de cara al antiguo robot, que puede ahora ser absorbido. El indicador en la parte superior derecha de la pantalla es el de advertencia de barrido. Si le localiza un Sentinel/Sentry, el indicador aparecerá lleno de motas. Tiene aproximadamente 5 segundos para desaparecer de vista antes de que se le agote la energía, 1 punto a la vez. Una vez agotadas toda la energía, queda absorbido totalmente y acaba la jugada.

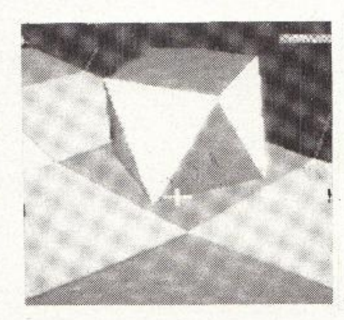
El volumen de energía total en el panorama permanece constante, de forma que si se absorbe 1 unidad de su energía, se colocará al azar un árbol en el panorama.

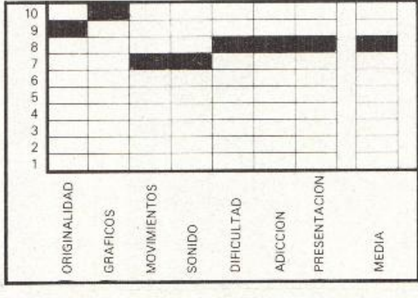
Si el indicador de advertencia de barrido sólo se encuentra medio lleno de motas, esto significará que el Sentinel/Sentry puede verle pero no puede ver el cuadro en el que permanece usted situado.











SENTINEL

SENTINEL es un juego sobrecogedor tanto por su temática como por los desoladores escenarios en los que transcurre la acción.

Podríamos calificar a SENTINEL como un simulador de una realidad futura: el control de un robot que nos trasmite la visión de otro mundo a través de un monitor de vídeo.

Los ESCENARIOS (10.000 en total) representa diferentes caras de un PLANETA cuyo paisaje es un enjambre de figuras geométricas.

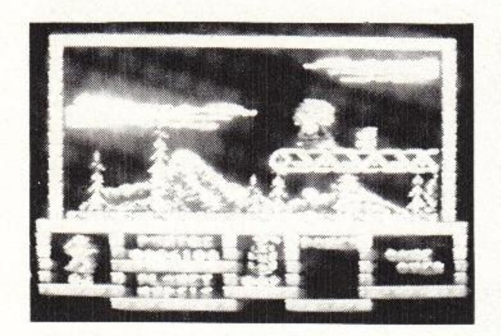
Hemos de descubrirnos ante el método de elaboración de los gráficos, así como también es de destacar la originalidad del tema y su ejecución.

La pantalla de presentación y los sonidos que amenizan y dan ambiente al juego también tienen una gran calidad.

SENTINEL es un juego que puede parecer en un principio complicado de manejar, no sólo por el número de teclas que utiliza sino por la cantidad de situaciones y posibilidades que ofrece; pero una vez conocido su manejo es apasionante.

Los movimientos son algo lentos, pero esto es totalmente disculpable dada la complejidad que entraña mover en pantalla unos paisajes tan elaborados.

GAME OVER



En una lejanísima Galaxia perdida en la inmensidad del Universo, una bellísima y malvada mujer había sometido con su inteligencia y su inmenso ejército de TERMINATORS a las cinco confederaciones de planetas situadas más allá de ALFA CENTAURI.

Todo el poder de GREMLA descansaba en la fidelidad casi religiosa de su lugarteniente ARKOS; sin embargo, la inteligencia de este MEGA-TERMINATOR de combate había evolucionado en los últimos ultraquarks de tal manera que, finalmente, abandonó a GREMLA y se unió a las tropas de la infantería sideral, que capitaneaba el general de estrella azul MC KILLER.

La aventura comienza...

Arkos conoce muy bien el poder de combate de los TERMINATORS de GREMLA.

Arkos es una máquina perfecta para destruirles.

Arkos sólo tiene una orden en su cerebro electrónico: ejecutar la misión.

ENEMIGOS Y PERSONAJES

Los planetas dominados por el poder de GREMLA se encuentran vigilados por un complejo centro de seguridad dirigido por el SISTEMA DE INTELIGENCIA SIDERAL, "NEXUS 2H07".

Ningún detalle escapa a su control.

Ninguna formación de tubo-cazas puede pasar inadvertida a su radar de partículas.

Nada es desconocido para NEXUS ZH07.

Si la presencia de NEXUS es determinante, no lo son menos las multitud de TERMINATORS que pululan por los planetas de GREMLA.

Todos ellos son máquinas creadas para el combate, la búsqueda y la destrucción.

GAME OVER es un juego de una acción desbordante: la velocidad con que se mueve su protagonista, la enorme fluidez de su disparo y el constante aparecer de enemigos por todas partes y a gran velocidad ponen en apuros al más templado jugador.

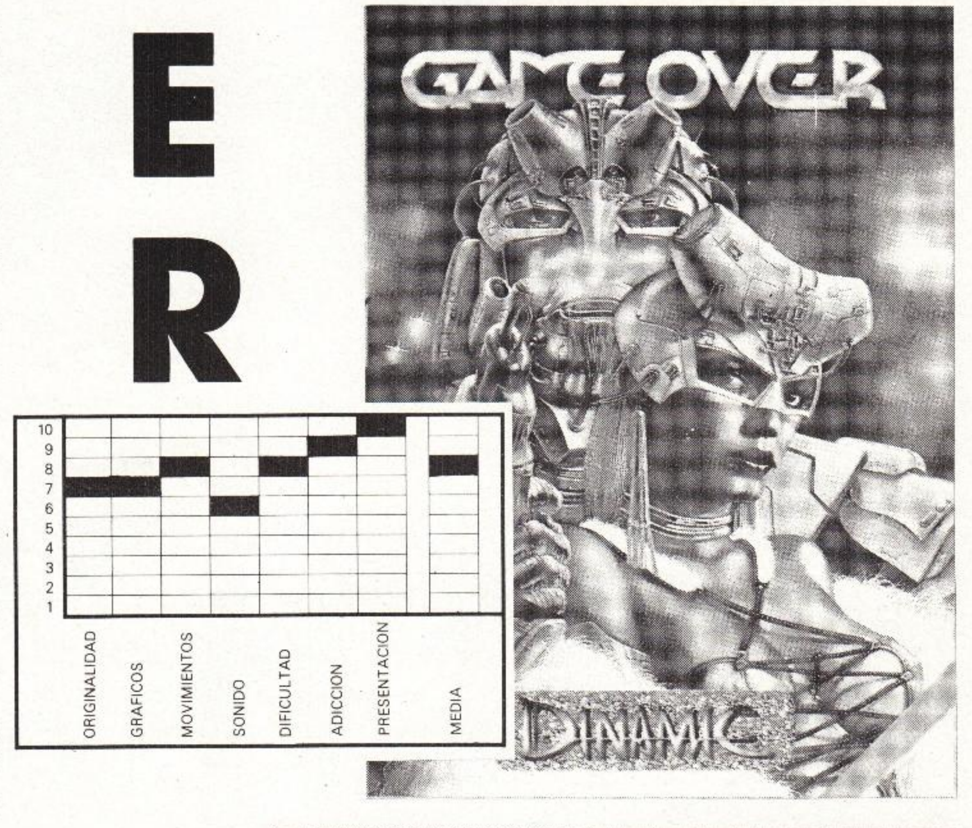
Los gráficos son bastantes variados y los movimientos rapidísimos así como la respuesta de "ARKOS" al joystick o al teclado.

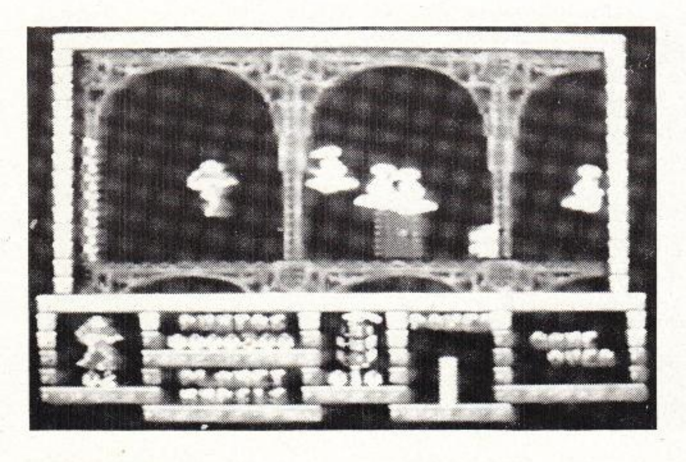
La pantalla de presentación constituye por sí misma una verdadera obra de arte, como ya viene siendo habitual en las producciones de DINAMIC.

Los sonidos amenizan los disparos e impactos que se realizan durante las dos fases o niveles de que se compone el juego.

El nivel de dificultad es muy elevado y requiere gran rapidez de reflejos y nervios de acero para ir superando cada pantalla.

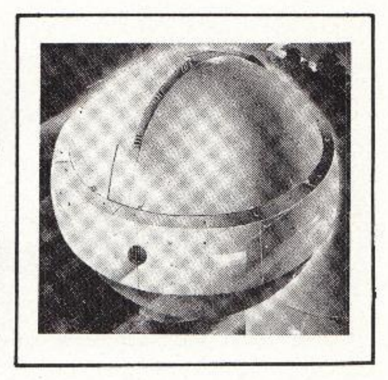
GAME OVER es en definitiva el juego ideal para aquellos que gusteis de las emociones fuertes y de los juegos de mucha acción.





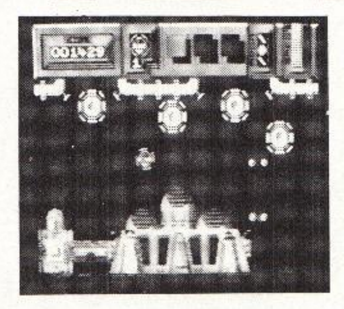


KINETIC

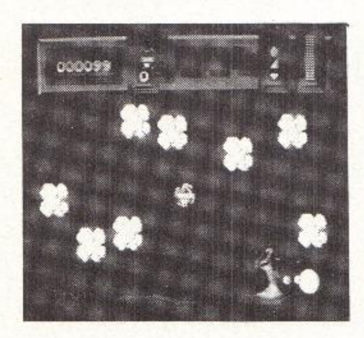


VIDEO-AVENTURA

Kinetik es un juego que presenta una gran calidad y variedad en sus gráficos, la respuesta al teclado también es rápida y los movimientos de sus protagonistas son ágiles, suaves y reflejan un trabajo costoso por parte de sus autores, que proporciona a Kinetik un aspecto muy agradable. Otro efecto destacable es que en la misma pantalla los marcadores están realizados en modo



1, mientras el juego está en mode 0. Los sonidos son escasos y se limitan a producir diferentes efectos sonoros en choques y demás lances del juego. La carga se realiza con un contador de bloques y ofrece una pantalla de presentación no demasiado elaborada. Kinetik es en líneas generales un juego muy elaborado y con una temática sumamente original y una gran variedad de situaciones y pantallas diferentes que le hacen bastante adictivo.



Por todo el universo las leyes del movimiento permanecen constantes, si se alteran o distorsionan estas leyes todo se convierte en caos. El moverse en el espacio se convierte en un reto a prueba de habilidad y destreza, y el evitar los objetos circundantes supone una batalla de tácticas.

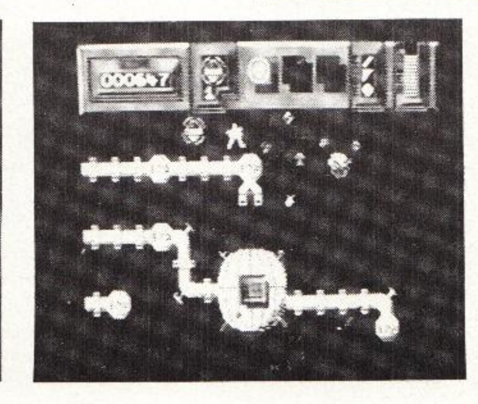
Y así es el mundo Kinético. Tu misión es restaurar las leyes a su equilibrio natural. Hay que llevar el mensaje de paz a las manos del kinemator, quien entonces ejercerá su control sobre las fuerzas del cosmos. Las tres letras que forman la palabra del mensaje se encuentran repartidas por las 43 zonas de Kinetik, y tu tienes que encontrarlas.

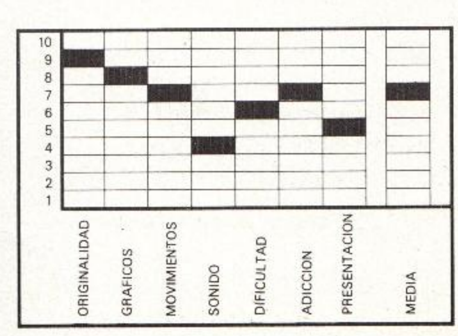
Tu hidronave esferoide se encuentra idealmente equipada para soportar las presiones de las fuerzas de gravedad que te harán frente, pero al iniciar el juego no hay forma de retarlas. Según se progresa, podrás ir recogiendo armamento, una diversidad de protección contra las fuerzas deflectora y un sistema teleportivo que servirá de protección contra las fuerzas de kinetik o de arma contra la multitud de sus habitantes. Para recoger el equipo sólo hay que moverse sobre él y éste quedará instalado en la caja de instrumentos entonces elegida (iluminada de rojo). Toda pieza de equipo que ya haya sido instalada en la caja será retirada y teleportada de nuevo a su lugar de origen.

EQUIPO

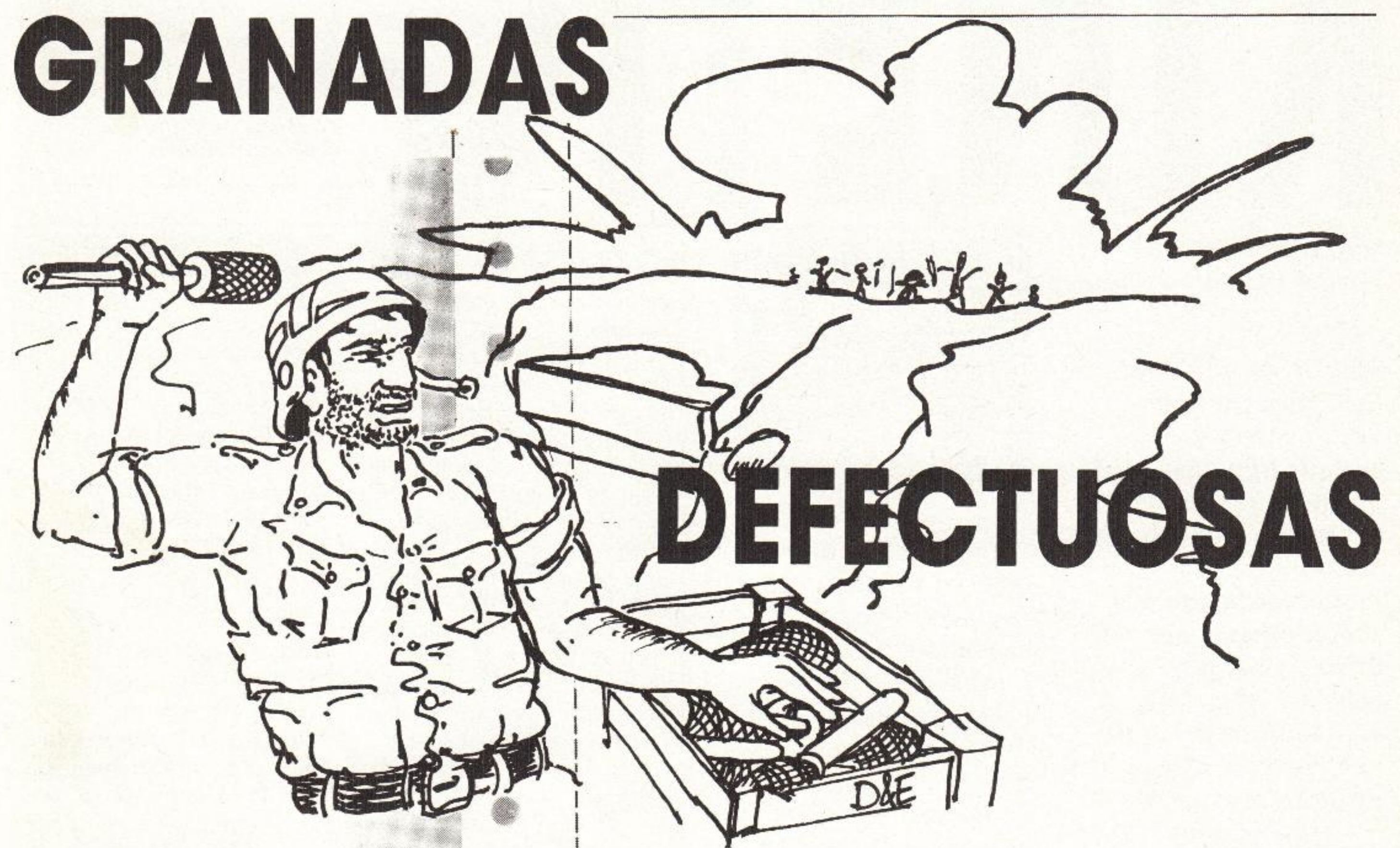
Pantalla Deflectora - Protege contra todos los habitantes, excepto el ladrón.
Dispersor Químico - Activado por la llave de accionamiento, este arma dispersa a los habitantes y sus ácidos super-corrosivos también hacen diente a través de los muros.
Impulsores de Control EZ Anula las fuerzas de gravedad, incluso las de fricción y salto en el planeta, lo cual permite el movimiento normal libre.
Impulsores Anti-Gravedad Sólo anulan la gravedad.
El Sistema Teleportivo Permite el teleporte a cualquier zona, excepto las de defensa central que conducen al Kinemator.







GRANADAS DEFECTUOSAS



Te encuentras aislado en una trinchera rodeado por el enemigo y posees una caja de granadas de mano, donde sabes que 5 de ellas son defectuosas, no obstante has de ir lanzándolas para conservar la vida aunque sólo momentáneamente.

Para lanzar las granadas tendrás que introducir en primer lugar la fila (1.5) y luego la columna (A-H).

Comentario del programa

Este es un juego donde se puede comprender fácilmente el funcionamiento de las tablas dimensionadas con DIM.

10-120. Redefinición de los gráficos.

120-260. Impresión de la pantalla de presentación en MODO 0.

```
10 REM ****   G R A N A D A S   ****
20 REM *** ANGEL GARCIA DELGADO ****
30 REM ******  MUNDO AMSTRAD  ******
40 '
50 ' -- REDEFINIR GRAFICOS --
60 '
70 SYMBOL AFTER 160
80 SYMBOL 160,0,63,112,111,207,192,134,1
42
90 SYMBOL 161,12,210,210,236,248,0,208,2
16
100 SYMBOL 162,0,110,110,110,0,59,59,27

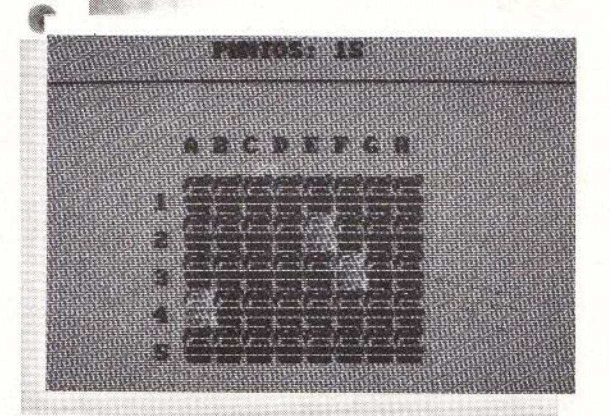
110 SYMBOL 163,0,238,238,238,0,220,220,2
16
120 '
130 ' -- PRESENTACION --
140 '
150 MODE 0:INK 0,24:INK 1,0:INK 2,6:INK
3,18
160 BORDER 24:PAPER 0:CLS
170 ORIGIN 300,200
180 FOR A=0 TO 2*PI STEP PI/20
190 PLOT 0,0,3:DRAWR COS(A)*800,SIN(A)*8
00
200 NEXT A
210 PRINT CHR$(22)+CHR$(1)
```

```
220 PEN 1
230 LOCATE 7,13:PRINT"GRANADAS"
240 FOR A=1 TO 1000 STEP 4:SOUND 1,A,1,1
5,0,0,31:NEXT A
250 INK 3,18,24
260 FOR A=1 TO 1000 STEP 5:SOUND 1,A,1,1
5:SOUND 2,A+10,1,15:SOUND 3,1000-A,1,15:
NEXT A
270 a$=CHR$(160)+CHR$(161)+CHR$(10)+CHR$
(8)+CHR$(8)+CHR$(162)+CHR$(163)
280 INK 3,18
290 '
300 ' --- INSTRUCCIONES ---
310 '
320 MODE 1
330 PEN 1
340 PRINT "  << LAS GRANADAS DEFECTUOSAS
 >>"
350 PRINT
360 PRINT
370 PRINT"     HAS DE IR LANZANDO GRANADA
S"
380 PRINT
390 PRINT"     DE LA CAJA. PERO EXITEN 5
QUE"
400 PRINT
410 PRINT"     SON DEFECTUOSAS, Y SI LAS"
420 PRINT
430 PRINT"     INTENTAS LANZAR TE ESTALLA
RAN"
440 PRINT
450 PRINT"     EN LAS MANOS (SON VERDES).
"
460 FOR A=1 TO 39 STEP 4:PEN 2:LOCATE A,
20:PRINT A$;:PEN 3:PRINT A$:NEXT A
470 INK 2,6,18:INK 3,18,6
480 IF INKEY$="" THEN 480
490 '
500 ' --- INICIALIZA ---
510 '
520 INK 2,6:INK 3,18
530 MODE 1:P=0:DIM M(8,5)
540 PEN 1
550 '
560 ' --- SITUA GRANADAS VERDES ---
570 '
580 FOR A=1 TO 5:M(1+INT(RND*8),1+INT(RN
D*5))=2
590 NEXT A
600 '
610 ' --- DIBUJA CELDILLAS ---
620 '
630 MODE 1
640 ORIGIN 0,200
650 PLOT 0,150,1:DRAWR 700,0
```

270-280. Se carga en A$ los caracteres necesarios para imprimir la granada. Se emplean también códigos de control para desplazar el cursor e imprimir cada carácter en su posición. El código 10 mueve el cursor hacia abajo, el código 8 lo mueve un carácter a la izquierda.

290-490. Impresión de las instrucciones del juego. Acompañada de un pequeño efecto de animación de tintas, en espera de la pulsación de una tecla para empezar.

500-560. Se fija el MODO 1 de pantalla con sus colores correspondientes, se ponen los puntos a cero y se dimensiona (con DIM) una tabla de



ocho elementos de anchura por 5 de altura.

560-590. Se sitúan mediante un bucle las 5 granadas defectuosas introduciendo un 2 en la tabla. Esto lo hacemos para luego al leer un 2 reconocer que se trata de una granada defectuosa.

600-790. Se activa el MODO 1 en pantalla. Se dibuja la rejilla y los nú-

Granadas

meros y letras de las celdillas. Para imprimir los números verticalmente se ha almacenado una cadena y luego se imprime con un bucle cada carácter de esta cadena.

800-900. Bucle principal. Se imprime el marcador de los puntos. Se pregunta el número de la fila y se comprueba si el valor no está comprendido en el margen, en este caso se vuelve a hacer la pregunta.

A continuación, el ordenador nos pide la columna en una variable alfanumérica. Se pasa a mayúsculas y se busca dentro de una cadena para obtener el número de columna a la que corresponde.

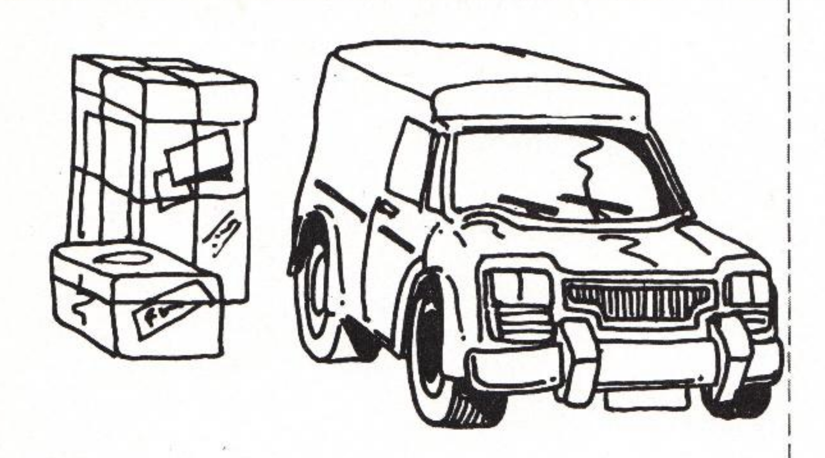
En la variable B se guarda el contenido de la celdilla seleccionada. Si B es 2 se salta a la rutina de explosión. Si B es 0 se suman 5 puntos, y se activa la celdilla a 1 para reconocer si se vuelve a hacer otra lectura de la misma celdilla, en caso afirmativo se preguntarán las coordenadas de nuevo.

En la línea 980 está la subrutina que dibuja la granada en la celdilla correspondiente, estando en la 990 la información para imprimir el marcador de los puntos.

1000-1120. Rutina de granada defectuosa. Sonido de fallo. Se muestra la localización de todas las demás granadas, revelando cuales eran las buenas y cuales las defectuosas. Luego se imprime el mensaje GAME OVER. Se hace una pausa y se salta a la zona de instrucciones.

```
660 PLOT 0,-150:DRAWR 700,0
670 WINDOW #1,12,35,9,19
680 PEN#1,1
690 LOCATE 12,7:PRINT"A B C D E F G H"
700 B$="1 2 3 4 5 "
710 FOR X=1 TO 10 STEP 2
720 PEN 1:LOCATE 10,X+9:PRINT MID$(B$,X,
1)
730 FOR Y=1 TO 16 STEP 2
740 PEN 1:LOCATE#1,Y,X
750 PRINT#1,A$
760 NEXT Y:NEXT X
770 WINDOW #2,10,40,24,24
780 PEN#2,1
790 P=0
800 '
810 ' --- BUCLE PRINCIPAL ---
820 '
830 GOSUB 990
840 SOUND 1,200,20,15
850 INPUT #2,"  FILA (1-5):";F
860 IF F<1 OR F>5 THEN SOUND 1,400,50:GO
TO 850
870 INPUT #2,"COLUMNA (A-H):";C$
880 C$=UPPER$(C$)
890 B$="ABCDEFGH"
900 C=0:FOR A=1 TO LEN(B$)
910 IF C$=MID$(B$,A,1) THEN C=A
920 NEXT A:IF C=0 THEN SOUND 1,400,50:GO
TO 870
930 B=M(C,F)
940 IF B=2 THEN PEN#1,3:GOSUB 980:SOUND
1,400,50:GOTO 1030
950 IF B=0 THEN M(C,F)=1:P=P+5:PEN#1,2:G
OSUB 980:FOR A=1 TO 1000 STEP 10:SOUND 1
,A,1,15:NEXT A
960 IF B=1 THEN SOUND 1,400,200:GOTO 850
970 GOTO 830
980 LOCATE #1,C*2-1,F*2-1:PRINT #1,A$:RE
TURN
990 LOCATE 14,2:PRINT "PUNTOS:";P:RETURN
1000 '
1010 ' --- GRANADA DEFECTUOSA ---
1020 '
1030 FOR A=400 TO 0 STEP -1:SOUND 1,A,1:
NEXT A
1040 FOR C=1 TO 8:FOR F=1 TO 5
1050 N=M(C,F):IF N=0 OR N=1 THEN PEN#1,2
1060 IF N=2 THEN PEN#1,3
1070 GOSUB 980
1080 SOUND 1,100,5
1090 FOR N=1 TO 50:NEXT N
1100 NEXT F:NEXT C
1110 PRINT #2," *** GAME OVER ***"
1120 FOR A=1 TO 5000:NEXT A:RUN 260
```

RECOGIDA



```
10 REM ******* RECOGIDA *******
20 '
30 REM * ANGEL GARCIA DELGADO *
40 '
50 REM ******* RECOGIDA *******
60 '
70 '
80 MODE 0
90 INK 0,0:INK 1,26:INK 2,7:INK 3,18
100 PAPER 0:BORDER 0
110 PRINT CHR$(22)+CHR$(1);
120 LOCATE 7,12
130 PRINT "RECOGIDA"
140 PRINT CHR$(22)+CHR$(0);
150 INK 2,7,0:INK 3,0,18
160 FOR a=1 TO 300:SOUND 1,a,1,15:NEXT a
170 INK 2,18,7:INK 3,7,18
180 FOR a=300 TO 1 STEP -2:SOUND 1,a,1:S
OUND 2,a*2,1:NEXT a
190 INK 2,7:INK 3,18
200 FOR a=1 TO 500:NEXT a
210 FOR a=0 TO 50:PRINT:FOR b=1 TO 50:NE
XT b:NEXT a
220 MODE 1
230 GOTO 300
240 PRINT TAB(20-LEN(a$)/2):FOR a=1 TO L
EN(a$)
250 b$=MID$(a$,a,1):PRINT b$;:IF b$<>" "
 THEN PRINT CHR$(7);
260 NEXT a
270 PRINT:PRINT:FOR a=1 TO 200:NEXT a:RE
TURN
280 '
290 '
300 REM *** INICIO DEL PROGRAMA ***
310 '
320 '
```

La época de la cosecha ha llegado. Los perales han dado su fruto y los comerciantes esperan impacientes la llegada de las primeras remesas de peras.

Si deseas que la competencia no te pise el terreno, tendrás que recoger todas las peras de la pantalla y llevarlas al camión antes de que éste se marche (cuando el tiempo sea cero).

Si consigues mantener el negocio, la cosecha de cada año será más próspera y tendrás que recoger más peras en menos tiempo.

Comentario del programa

RECOGIDA, es un programa donde el elemento de atracción fundamental es la velocidad de su desarrollo. El jugador ha de recoger las peras de la pantalla en un tiempo límite. La dificul-

tad se incrementa a medida que se completa una pantalla.

10-270. Presentación. Este bloque incluye:

— preparación de la pantalla en MODO 0. 20 columnas de ancho de impresión y uso de 16 colores.
— Impresión del título.
— Efectos de color y sonido.
— Borrador del título con desplazamiento de pantalla.

280-430. Instrucciones. Realizadas en el MODO 1, con el empleo de una rutina de centrado. La pausa de la línea 430 nos permite leer las instrucciones.

440-560. Inicialización del juego. Las variables de la fase, los puntos y el tiempo se cargan con su valor. Se dibuja el escenario, se imprimen los marcadores y se crea la ventana donde se desarrollará la acción.

570-600. Se inicializan las fases. Se sitúan las peras. El tiempo toma el valor máximo y las coordenadas son puestas en la posición de salida.

610-730. Bucle principal. Se lee el teclado, actuando en consecuencia. Borramos nuestro muñeco. Obtenemos el código del carácter de la posición donde se va a imprimir y vemos si se trata de una pera, gracias al código ASCII = 160, luego comprobamos si tenemos ya todas las peras para pasar de fase. Imprimimos

```
330 A$="<<< RECOGIDA >>>":GOSUB 240
340 PRINT:A$="RECOGE LAS PERAS DEL HUERT
O PARA":GOSUB 240
350 A$="CARGARLAS EN EL CAMION, ANTES DE
 QUE":GOSUB 240
360 A$="ESTE SE MARCHE.":GOSUB 240
370 PRINT:A$="CONTROLES DEL JUEGO:":GOSU
B 240
380 A$="Q. ARRIBA":GOSUB 240
390 A$="A. ABAJO":GOSUB 240
400 A$="O. IZQUIERDA":GOSUB 240
410 A$="P. DERECHA":GOSUB 240
420 PRINT:A$="PULSA -S- PARA EMPEZAR":GO
SUB 240
430 IF INKEY(60)<>0 THEN GOTO 430
440 P=0:T=250:F=1
450 MODE 1
460 INK 0,0:INK 1,26:INK 2,9:INK 3,24
470 BORDER 0:PAPER 0
480 PEN 1
490 WINDOW 1,40,7,25
500 GOSUB 990:GOSUB 1090
510 PRINT #1,"    PUNTOS:";TAB(20);"    TI
EMPO:"
520 A$="":B$=""
530 FOR A=1 TO 40
540 A$=A$+CHR$(166):B$=B$+CHR$(165)
550 NEXT A
560 PRINT #1,A$:PRINT #1,B$;"   ";CHR$(16
1);CHR$(162);CHR$(163);CHR$(164):PRINT #
1,A$
570 CLS:GOSUB 880
580 PEN 3
590 T=300
600 X=5:Y=5:XB=5:YB=5:R=0
610 IF INKEY(67)=0 THEN X=X-1:IF X<1 THE
N X=1
620 IF INKEY(69)=0 THEN X=X+1:IF X>18 TH
EN X=18
630 IF INKEY(34)=0 THEN Y=Y-1:IF Y<1 THE
N Y=1
640 IF INKEY(27)=0 THEN Y=Y+1:IF Y>40 TH
EN Y=40
650 LOCATE YB,XB:PRINT" "
660 LOCATE Y,X:CALL 30000:C=PEEK(29999)
670 IF C=160 THEN R=R+1:SOUND 1,500,3:P=
P+5:LOCATE #1,11,1:PRINT#1,P
680 IF R>9+5*F THEN R=0:F=F+1:P=P+T:FOR
A=1 TO 1000 STEP 3:SOUND 1,A,1:NEXT A:GO
TO 570
690 PRINT CHR$(248+RND*3)
700 T=T-1:LOCATE #1,31,1:PRINT#1,USING "
###";T
710 IF T<1 THEN GOTO 740
720 XB=X:YB=Y
```

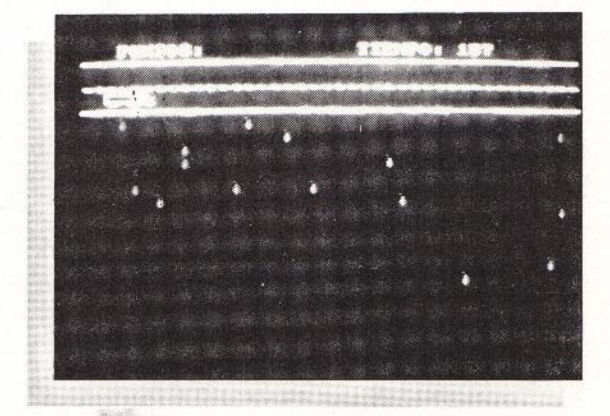
```
730 GOTO 610
740 B$=" "+CHR$(161)+CHR$(162)+CHR$(163)
+CHR$(164)
750 FOR A=3 TO 36
760 LOCATE #1,A,5:PRINT #1,B$
770 SOUND 1,1000,1:FOR B=1 TO 50:NEXT B
780 NEXT A
790 MODE 0
800 PEN 2
810 LOCATE 6,7:PRINT"GAME OVER"
820 PEN 1
830 LOCATE 5,12:PRINT"PUNTOS:";USING "##
##";P
840 INK 2,11,0
850 FOR A=1 TO 4000 STEP 10:SOUND 1,A,1:
NEXT A
860 FOR A=1 TO 10000:NEXT A
870 RUN 220
880 FOR A=1 TO 10+5*F
890 X=1+INT(17*RND):Y=1+INT(RND*39)
900 LOCATE Y,X:CALL 30000:IF PEEK(29999)
<>32 THEN GOTO 890
910 PEN 2:PRINT CHR$(160)
920 PRINT CHR$(7);
930 NEXT A
940 RETURN
950 END
960 '
970 '
980 '
990 REM ** CODE SCREEN$ **
1000 '
1010 '
1020 DATA 205,96,187,50,47,117,201
1030 RESTORE 1020
1040 FOR a=30000 TO 30006:READ b:POKE a,
b:NEXT a
1050 RETURN
1060 '
1070 REM *****************
1080 '
1090 SYMBOL AFTER 160
1100 SYMBOL 160,96,144,56,56,124,124,124
,56
1110 SYMBOL 161,255,128,128,128,128,255,
91,27
1120 SYMBOL 162,255,0,0,0,0,255,64,0
1130 SYMBOL 163,251,11,11,11,11,251,222,
216
1140 SYMBOL 164,224,176,152,143,255,255,
93,28
1150 SYMBOL 165,0,0,0,0,126,63,0,0
1160 SYMBOL 166,0,0,0,255,255,0,0,0
1170 RETURN
```

nuestro muñeco, animado con los caracteres incluídos en el banco gráfico original.

Luego decrementamos en 1 el tiempo, lo imprimimos y comprobamos si es cero. Si lo es se salta al final, si no se repite el bucle.

740-870. El camión se marcha. Se imprime el mensaje de final y tras una pausa se lanza de nuevo el programa.

880-960. Subrutina que pinta las peras en pantalla. Calcula unas coordenadas al azar. Comprueba si hay pera. Si existe se calculan de nuevo las coordenadas

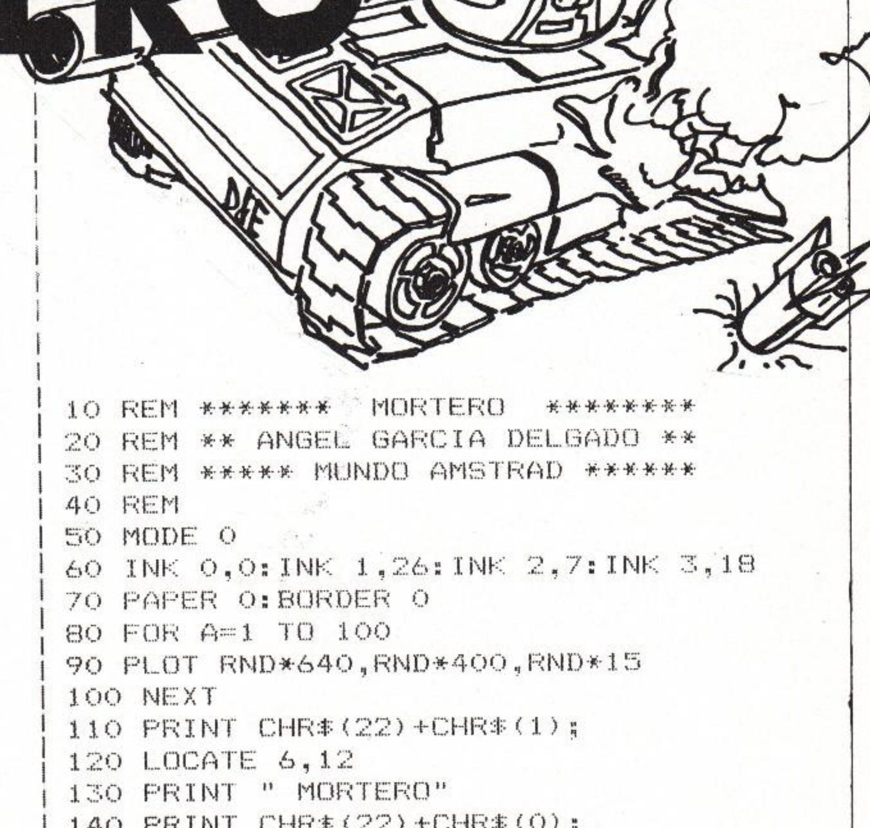


y se repite el proceso, sino, se dibuja la pera.

970-1060. Subrutina que introduce en la memoria los DATAS de la rutina en C/M que nos permite averiguar el carácter que hay en la posición del cursor.

1070-1170. Subrutina de redefinición de gráficos. Aquí se cambian varios gráficos por las definiciones del camión, las líneas de la carretera y la pera.

MORTERO



Fuerzas de tanques enemigos se dirigen hacia el punto estratégico que defiendes, "las montañas CABRA-LOCA". Has de impedir que franqueen este obstáculo rocoso. Para hacerlo dispones de un mortero de gran alcancer que te permitirá destruir los tanques.

Debes dar el ángulo y el impulso correctos para fijar la trayectoria que debe seguir la bola. El impulso no debe sobrepasar la cantidad indicada en el marcador de impulso. Si éste llega a ser cero, nuestro mortero quedará inutilizado y el enemigo tomará nuestra posición.

Comentario del programa

Se trata de un programa donde se han dejado apartadas las rutinas de acción para pasar a procesar una serie de cálculos, que decidirán la trayectoria parabólica que describirá nuestro obús.

10-210. Presentación. Se activa el MODO 0 de pantalla, se les da el color a las tintas y se dibuja un fondo estrellado. A continuación, en modo transparente para conservar las estrellas del fondo al imprimir el título. Tras unos efectos sonoros, se limpia la pantalla para dar paso a las instrucciones.

220-400. Con la pantalla en MODO 1, son impresas las instrucciones con una rutina de centrado. Y se espera la pulsación de la tecla "S" para continuar.

410-490. Se redefinen los cuatro gráficos que nos son necesarios.

500-970. Bucle principal del juego. Incluye la inicialización de las variables de impulso IM y los aciertos AC. Se establecen los colores de las tintas y se pasa de nuevo a MODO 0. Luego se define una función que emplea un parámetro, y que genera un número al azar entero de 0-(n-1), donde "n" es el valor del parámetro introducido. El origen de

```
10 REM *******  MORTERO  ********
20 REM ** ANGEL GARCIA DELGADO **
30 REM ***** MUNDO AMSTRAD ******
40 REM
50 MODE 0
60 INK 0,0:INK 1,26:INK 2,7:INK 3,18
70 PAPER 0:BORDER 0
80 FOR A=1 TO 100
90 PLOT RND*640,RND*400,RND*15
100 NEXT
110 PRINT CHR$(22)+CHR$(1);
120 LOCATE 6,12
130 PRINT " MORTERO"
140 PRINT CHR$(22)+CHR$(0);
150 INK 2,7,0:INK 3,0,18
160 FOR a=1 TO 300:SOUND 1,a,1,15:NEXT a
170 INK 2,18,7:INK 3,7,18
180 FOR a=300 TO 1 STEP -2:SOUND 1,a,1:S
OUND 2,a*2,1:NEXT a
190 INK 2,7:INK 3,18
200 FOR a=1 TO 500:NEXT a
210 FOR a=0 TO 50:PRINT:FOR b=1 TO 50:NE
XT b:NEXT a
220 MODE 1
230 GOTO 290
240 PRINT TAB(20-LEN(a$)/2):FOR a=1 TO L
EN(a$)
250 b$=MID$(a$,a,1):PRINT b$;:IF b$<>" "
 THEN SOUND 1,A,1,15
260 NEXT a
270 PRINT:PRINT:FOR a=1 TO 200:NEXT a:RE
TURN
280 '
290 REM *** INICIO DEL PROGRAMA ***
300 '
310 A$="<< MORTERO >>":GOSUB 240
320 PRINT:A$="DESTRUYE UN TANQUE QUE SE
ENCUENTRA":GOSUB 240
330 A$="AL OTRO LADO DE LA CADENA DE MON
TES.":GOSUB 240
```

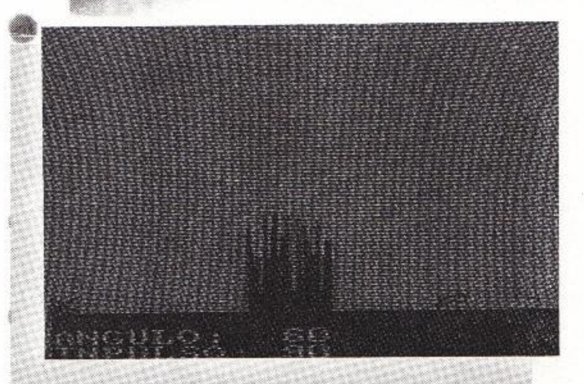
```
340 PRINT:A$="TU PUEDES REGULAR EL ANGUL
O DE TIRO":GOSUB 240
350 A$="Y EL IMPULSO (VELOCIDAD INICIAL)
":GOSUB 240
360 A$="DE TU UNIDAD ARTILLERA":GOSUB 24
0
370 PRINT:A$="POR CADA TANQUE DESTRUIDO
SE TE":GOSUB 240
380 A$="AUMENTARA EN 100 UNIDADES EL INP
ULSO":GOSUB 240
390 PRINT:A$="PULSA -S- PARA EMPEZAR":GO
SUB 240
400 IF INKEY(60)<>0 THEN 400
410 '
420 ' * REDEFINICION GRAFICOS *
430 '
440 SYMBOL AFTER 32
450 SYMBOL 91,1,7,127,7,127,255,255,127
460 SYMBOL 93,128,224,224,224,254,255,25
5,254
470 SYMBOL 92,218,173,189,81,127,103,188
,24
480 SYMBOL 64,129,66,36,152,104,102,233,
150
490 '
500 ' * INICIALIZACION *
510 '
520 IM=300:AC=0
530 INK 0,23:PAPER 0:BORDER 23:INK 1,0:I
NK 2,9:INK 3,2:INK 4,3:INK 5,26:PEN 1:MO
DE 0
540 DEF FN R(X)=INT(RND*X):ORIGIN 0,0
550 LOCATE 1,22:PEN 4:PRINT"@"
560 FOR A=0 TO 47:PLOT 0,A,2
570 DRAWR 650,0
580 NEXT A
590 WINDOW #2,1,20,24,25:PAPER #2,2:PEN
#2,5
600 WINDOW #1,1,20,1,2:PEN #1,5
610 GOSUB 1040
620 FOR Z=250 TO 350 STEP 4
630 PLOT Z,47:DRAWR 0,FN R(100)
640 NEXT Z
650 X=19-FN R(8)
660 LOCATE X,22:PEN 3:PRINT"[]"
670 IF IM<1 THEN GOTO 1010
680 CALL &BB03
690 CLS #2:INPUT #2,"ANGULO:  ",ANG
700 INPUT #2,"INPULSO: ",I:IF I<0 OR I>I
M THEN SOUND 1,100:GOTO 700
710 IM=IM-I
720 X=1:Y=200
730 ENT 1,10,1,1,20,-1,1
740 DEG:SOUND 1,500,30,15,0,1,31
750 VX=I*COS(ANG):VY=I*SIN(ANG)
```

coordenadas gráficas se sitúa en la coordenada 0,0 (esquina inferior izquierda). Se dibuja nuestro cañón, el suelo, se imprimen los marcadores, definiendo previamente un par de ventanas (la segunda usada para la entrada de nuestros datos). Luego de forma aleatoria son dibujadas las montañas, y es situado el tanque enemigo.

Entramos ya realmente en lo que es el bucle del juego (línea 670). La primera comprobación que hacemos es si queda impulso. Si este es cero se salta al final. Sino, limpiamos el buffer del teclado, y preguntamos el ángulo y el impulso. El ángulo puede ser cualquier número, pero el impulso está limitado entre cero y el impulso que tenemos, de forma que si se introduce un valor erróneo, un sonido lo indica y se vuelve a hacer la pregunta.

Se resta el impulso elegido del total. Se inicializan las coordenadas del obús, y se fija la entonación 1. Con DEG pasamos los cálculos a grados (en vez de radianes). Sonido de lanzamiento.

A partir de la línea 750 es donde se realiza el movimiento del



obús con sus correspondientes comprobaciones de choque.

Lo primero que se hace es empleando la trigonometría, calcular los vectores incremento para X e Y. Se entra en el bucle de movimiento, se borra la posición anterior del obús, y en las líneas 780 y 790 se calcula en función de T (simulacro de tiempo) la coordenada y empleando la fórmula del tiro parabólico. En la 800 la variable TI toma el valor del color del punto donde se

Mortero

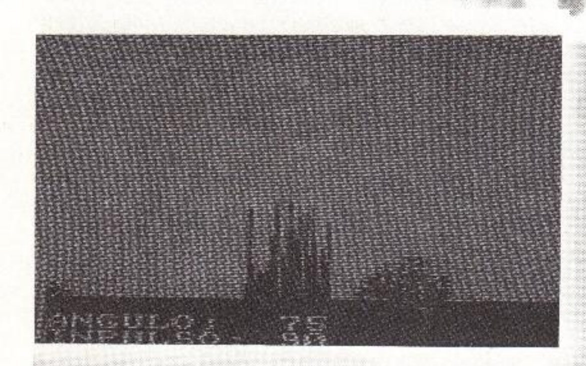
va a imprimir el obús, se hace un sonido, una pausa y se pasa a las comprobaciones de choque. Si TI = 2 se choca contra el paisaje, y se salta a la 880.

Si TI = 3 el choque es con el tanque y el salto es a la línea 980. Luego el NEXT continúa todo el proceso.

En la 880 se encuentra la rutina de explosión, que dentro de su sencillez produce un efecto batante bueno. Lo que hago es activar el modo transparente y el modo gráfico con TAG y se fija el origen de coordenadas en la posición. A continuación en un bucle se imprimen con desplazamientos al azar, asteriscos, que al irse incrementando el valor del bucle, cada vez alcazan un área más grande.

En la línea 980, continúa el programa al acertar al enemigo. Sonido de acierto. Se incrementa el contador de impactos, se imprimen los marcadores, se suma 100 al impulso y se dibuja la explosión del tanque.

1000-1020. Sonidos de final y salto a GAME OVER.



1030-1050. Subrutina de impresión de los marcadores.

1066-1180. GAME OVER. El juego finaliza, se ofrece la puntuación y la opción de empezar de nuevo a acabar, reinicializar el ordenador. (Ojo con esta parte del programa, grábalo antes de usar esta opción o perderás el programa).

```
760 FOR Z=8 TO 650 STEP 4
770 PLOT X,Y,0
780 T=Z/VX:X=Z+16
790 Y=55+VY*T-4.9*T*T
800 TI=TEST (X,Y)
810 PLOT X,Y,1
820 SOUND 2,ABS(Y*2),1,13
830 FOR V=1 TO 30:NEXT V
840 IF TI=2 THEN GOTO 880:REM -- CHOQUE
CON PAISAJE --
850 IF TI=3 THEN GOTO 980:REM --    ''
CON TANQUE --
860 IF Y<8 THEN GOTO 880
870 NEXT Z
880 ORIGIN X,Y
890 PRINT CHR$(23);CHR$(1):TAG:FOR N=1 T
O 50
900 MOVE RND*N,RND*N:PRINT"*";
910 MOVE -RND*N,RND*N:PRINT"*";
920 REM MOVE -RND*N,-RND*N:PRINT"*";
930 REM MOVE RND*N,-RND*N:PRINT"*";
940 SOUND 3,A*5,3,15,0,1,31
950 NEXT N:TAGOFF:PRINT CHR$(23);CHR$(0)
;
960 FOR N=1 TO 2000:NEXT N
970 GOTO 530
980 SOUND 3,50,30,15,0,1:AC=AC+1:GOSUB 1
040
990 IM=IM+100
1000 GOTO 880
1010 FOR A=1 TO 20:SOUND 3,50,30,15,0,1:
NEXT A
1020 GOTO 1050
1030 LOCATE 1,1:END
1040 CLS #1:PRINT#1,"IMPULSO:  ";USING "
###";IM:PRINT#1,"IMPACTOS: ";USING "###"
;AC:RETURN
1050 REM
1060 '
1070 REM *** GAME OVER ***
1080 '
1090 FOR A=1 TO 20:FOR B=100 TO 200 STEP
 A:SOUND 1,B,1,15,0,0,31:NEXT B:NEXT A
1100 INK 0,0:INK 1,26:INK 2,15,0:PAPER 0
:BORDER 0:PEN 1:MODE 0
1110 PEN 2:LOCATE 6,7:PRINT"GAME OVER"
1120 PEN 1:LOCATE 4,12:PRINT"PUNTOS:";US
ING "######";P
1130 PEN 3:INK 3,15:LOCATE 5,17:PRINT "O
TRA? (S/N)"
1140 FOR A=1 TO 50:A$=INKEY$:NEXT A
1150 PEN 1
1160 IF INKEY(60)=0 THEN RUN 220
1170 IF INKEY(46)=0 THEN CALL 0
1180 GOTO 1160
```

ARCHIVO NUMERICO

```
10 ' ********************************
20 ' **                            **
30 ' **      ARCHIVO NUMERICO      **
40 ' **                            **
50 ' **                            **
60 ' ********************************
70 '
80 '
90 ' ********************************
100 '          MUNDO-AMSTRAD
110 '********************************
120 '
130 '
140 '          PRESENTACION
150 '
160 '
170 MODE 0:BORDER 1:LOCATE 1,12:PEN 3:PR
INT"  ARCHIVA  NUMEROS  ":FOR t=10 TO 50
0 STEP 10:SOUND 5,t,10:NEXT
180 '
190 '     COMENTARIOS PRELIMINARES
200 '
210 MODE 1:PEN 1:PRINT:PRINT:PRINT"     E
n este archiva numeros podras "
220 PRINT  "  almacenar tantas cantidade
s como "
230 PRINT  "  te permita la memoria de t
u orde-"
240 PRINT  "  nador. Estas cantidades no
 pueden"
250 PRINT  "  tener mas de seis cifras c
ada uná, "
260 PRINT  "  es decir, el numero mayor
almace- "
270 PRINT " nable es 999999. El archivo
 viene"
280 PRINT      "  presentado en paginas d
e 30 datos"
290 PRINT      "  los cuales pueden ser m
odificados"
300 PRINT      "  tantas veces como se pr
ecise antes"
310 PRINT           "  e incluso despues d
e relizar cual-"
320 PRINT           "  quier tipo de calcu
lo con ellos."
330 PRINT :PRINT              "    Como ej
emplo de calculo hemos"
340 PRINT               "  puesto una esta
distica de errores,"
350 PRINT               "  si bien tu pued
es introducir otros."
360 LOCATE 1,22:PRINT"      PULSA UNA TEC
LA PARA COMENZAR"
370 IF INKEY$="" THEN 370 ELSE CLS
380 '
390 ' INTRODUCCION DEL NUMERO DE DATOS
400 '
410 PRINT "NUM. DE DATOS A INTRODUCIR
";
420 xj=28:yj=1:bu=0:GOSUB 1330
430 IF VAL(a$)=0 OR VAL(a$)>7500 THEN LO
CATE 28,1:PRINT "          ";:PRINT CHR$
(7);:GOTO 420
440 '
450 ' INTRODUCCION DE DATOS NUMERICOS
460 '
```

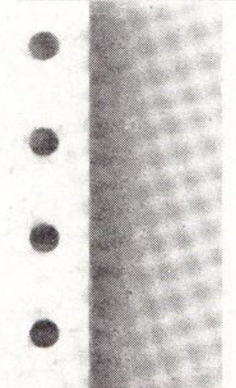
Este es un programa de utilidad que permite almacenar en tu Amstrad hasta 7.500 números, presentados en pantalla en orden y en columnas para su comprobación y control.

Estos datos numéricos son almacenados en una matriz y pueden ser recuperados para realizar con ellos cualquier tipo de operaciones. Como ejemplo hemos introducido un cálculo estadístico de errores aunque podría ser cualquier otra operación definida en BASIC por el usuario. Los números contenidos en este archivo pueden ser modificados en cualquier momento con sólo acudir a la página que los contiene y teclear su número de orden, después de haber pulsado la letra "M". La modificación se aloja en el mismo lugar que ocupaba el dato antiguo.

Explicación del listado

En la línea 170 tenemos un pequeñísima rutina que





se encarga de poner en pantalla, durante un corto espacio de tiempo, el rótulo que da título al programa.

Entre las líneas 210 y 370 se encuentran las instrucciones que aparecen en la pantalla antes de comenzar a introducir datos para que sepamos como hacerlo.

Entre las líneas 410 y 430 se solicita el número total de

Archivo numérico

datos que va a contener nuestro archivo para dimensionar la matriz que los contenga (línea 470).

Las líneas comprendidas entre la 480 y la 540 gestionan la introducción de cada dato hasta completar el número que hayamos determinado como final del archivo (línea 530). A partir de la línea 580 tenemos la rutina de salto de página que se encarga de presentar en pantalla los números en páginas de 30 datos (líneas 620-700) y en caso de que no quepan todos en una sola página la línea 700 se encarga de llamar a la rutina de comandos de supermisión. Cuando nuestro archivo alcanza el último dato, la línea 710 hará aparecer un mensaje advirtiéndonos que no hay más datos, asímismo las líneas 600 y 610 se encargan de bloquear el avance o el retroceso de páginas cuando estamos en la primera o en la última respectivamente. La rutina de comandos de supervisión comienza en la línea 760 y se encarga primeramente de mostrar en pantalla las distintas opciones que podemos utilizar (líneas 760-800) y posteriormente hace que el programa salte a las subrutinas que gestionen cada opción (líneas 800-850). En la línea 890 comienza la rutina de modificaciones que se encarga de hacer correcciones en los datos una vez introducidos y presentados en pantalla: la línea 900 nos pide el número del dato que queremos modificar, y si éste está en la pantalla, da paso a la línea 910 que nos presenta el contenido de dicho dato y la 920 nos pide el nuevo valor a introducir.

Una vez que indicamos a nuestro ordenador que los datos son correctos, pulsando la letra "C", el programa

```
470 d=VAL(a$):DIM a(d)
480 xj=16:yj=2:bu=0:FOR I=1 TO D
490 PRINT:PRINT "DATO NUMERO";I;"=";
500 GOSUB 1330
510 a(i)=VAL(a$)
520 yj=yj+1
530 IF I=D THEN PRINT:PRINT:PRINT"TODOS
LOS DATOS HAN SIDO INTRODUCIDOS":PRINT C
HR$(7);:FOR H=1 TO 800:NEXT H
540 NEXT
550 '
560 '              SALTO DE PAGINA
570 '
580 CLS
590 NP=1
600 IF NP=0 THEN PRINT"*****  NO HAY MEN
OS PAGINAS *****":NP=NP+1
610 IF NP*30-D>=30 THEN PRINT"***** NO H
AY MAS PAGINAS *****":NP=NP-1
620 Z=(NP-1)*30
630 FOR I=Z+1 TO D
640 LOCATE 1,2:PRINT"NUM. DE DATOS ";D
650 X=INT(((I-6)/10) MOD 30)*12-(NP-1)*3
6
660 IF x<0 THEN x=0
670 Y=((I-1) MOD 10)+4
680 LOCATE 23,2:PRINT"PAG. NUM.";NP
690 LOCATE x+1,y:PRINT i;"-";a(I);:b$=ST
R$(a(i)):PRINT STRING$(7-LEN(b$),32)
700 IF I=Z+30 THEN GOTO 760
710 IF I=D THEN LOCATE 5,16:PRINT"*** FI
N DE LOS DATOS ***"
720 NEXT
730 '
740 '       COMANDOS DE SUPERVISION
750 '
760 LOCATE 1,16:PRINT"PULSE + PARA PAG.
SIGUIENTE     "
770 LOCATE 1,18:PRINT"PULSE - PARA PAG.
ANTERIOR"
780 LOCATE 1,20:PRINT"PULSE M PARA MODIF
ICAR DATOS";:PRINT STRING$(12,32);
790 LOCATE 1,22:PRINT"PULSE C SI LOS DAT
OS SON CORRECTOS      "
800 A$=INKEY$
810 IF A$="+" THEN CLS:PRINT CHR$(7);:NP
=NP+1:GOTO 600
820 IF A$="-" THEN CLS:PRINT CHR$(7);:NP
=NP-1:GOTO 600
830 IF A$="M" OR a$="m" THEN PRINT CHR$(
7);:GOTO 890
840 IF A$="C" OR a$="c" THEN CLS:PRINT C
HR$(7);:GOTO 970
850 GOTO 800
860 '
870 '    RUTINA DE MODIFICACIONES
880 '
890 LOCATE 1,22:PRINT STRING$(36,32)
900 LOCATE 1,20:INPUT "INDIQUE NUM. DE D
ATO A MODIFICAR";I:IF I>NP*30 OR I<NP*30
-29 OR I>D THEN PRINT CHR$(7);:LOCATE 1,
22:PRINT"EL DATO NUM.";I;"NO ESTA EN PAN
TALLA":FOR V=1 TO 200:PRINT CHR$(7);:NEX
T:GOTO 760
910 PRINT CHR$(7);:LOCATE 1,20:PRINT"EL
DATO NR";I;"CONTIENE EL VALOR ";A(I);:PR
INT STRING$ (4,32)
```

```
920 LOCATE 1,22:INPUT "INTRODUCCA EL NUE
VO VALOR ";A(I):IF A(I)>999999 THEN PRIN
T CHR$(7);:LOCATE 1,22:PRINT STRING$(40,
32);:GOTO 920
930 GOTO 600
940 '
950 '   SALTO A LISTADOS O A CALCULOS
960 '
970 CLS:PRINT CHR$(7);:PRINT "QUE CALCUL
O REALIZARA CON SUS DATOS:":PRINT:PRINT"
-PULSE R PARA REVISAR DATOS "
980 PRINT:PRINT"-PULSE G PARA ESTADISTIC
A DE ERRORES"
990 B$=INKEY$
1000 IF B$="G" OR b$="g" THEN CLS:GOTO 1
060
1010 IF B$="R" OR b$="r" THEN CLS:GOTO 6
00
1020 GOTO 990
1030 '
1040 '          CALCULO DE ERRORES
1050 '
1060 PRINT CHR$(7):PRINT:PRINT"*** CALCU
LO ESTADISTICO DE ERRORES ***"
1070 FOR I=1 TO d
1080 s=s+a(i)
1090 NEXT
1100 m=s/d
1110 FOR i=1 TO d
1120 s2=s2+(a(i)-M)^2
1130 NEXT
1140 e=SQR(s2/(d*(d-1)))
1150 LOCATE 3,6:PRINT"MEDIA=";m
1160 LOCATE 3,9:PRINT"ERROR ABSOLUTO=";e
1170 er=e/m
1180 LOCATE 3,12:PRINT"ERROR RELATIVO=";
er
1190 te=er/100
1200 LOCATE 3,15:PRINT" % DE ERROR=";te
1210 '
1220 'NUEVOS CALCULOS O FIN DE PROGRAMA

1230 '
1240 LOCATE 1,20:PRINT"DESEA REALIZAR AL
GUN CALCULO MAS (S/N)"
1250 A$=INKEY$
1260 IF a$="s" OR a$="S" THEN s=0:s2=0:C
LS:GOTO 970
1270 IF a$="n" OR a$="N" THEN 1290
1280 GOTO 1250
1290 CLS:END
1300 '
1310 '    RUTINA DE INTRODUCCION
1320 '
1330 xu=xj:yu=yj:bu=0:a$=""
1340 x$=INKEY$: IF x$="" THEN 1340
1350 IF x$=CHR$(13) OR  bu>5 THEN LOCATE
 xu+1,yu:PRINT " ";:RETURN
1360 IF x$=CHR$(127) AND bu>0 THEN hh=LE
N(a$):a$=LEFT$(a$,hh-1):bu=bu-1:LOCATE x
u,yu:PRINT"_ ":xu=xu-1:GOTO 1340 ELSE IF
 x$=CHR$(127) THEN 1340
1370 a$=a$+x$:xu=xu+1:bu=bu+1:LOCATE xu,
yu:PRINT x$;"_ ";
1380 GOTO 1340
```

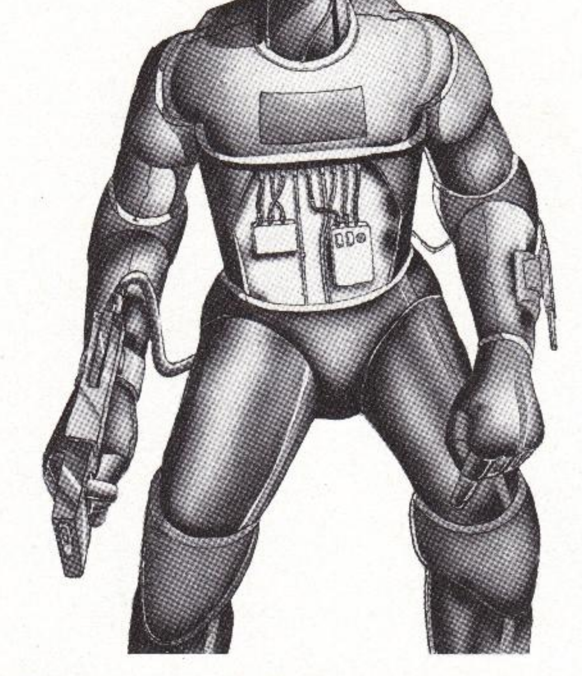
salta al menú de cálculo (línea 970) donde únicamente hemos introducido las opciones regresar a comprobación de datos o realizar un cálculo estadístico de errores (líneas 1000 y 1010).

En la línea 1060 comienza el cálculo de errores y finaliza en la línea 1200. Como ya hemos dicho pueden realizarse otros tipos de operaciones con sólo incluir en el programa unas líneas semejantes a la (980 y 1000) que hagan saltar la ejecución del listado a cualquier otra rutina que programemos en BASIC.

Entre las líneas 1240 y 1290 tenemos las opciones de fin de programa o regreso a nuevos cálculos.

Finalmente entre las líneas 1330 y 1380 se encuentra una rutina de introducción o captación de datos desde el teclado, que en base a unas coordenadas das "XJ" e "YJ" presenta en pantalla cada tecla que pulsemos guardando el contenido de la cadena impresa, en la variable "a$" y regresando a la línea siguiente a la que fue llamada dicha rutina de impresión. Como vereis ésta rutina puede ser utilizada en otros programas en los que necesiteis captar datos desde el teclado.

PROXIMAS SECCIONES



CORREO DEL LECTOR

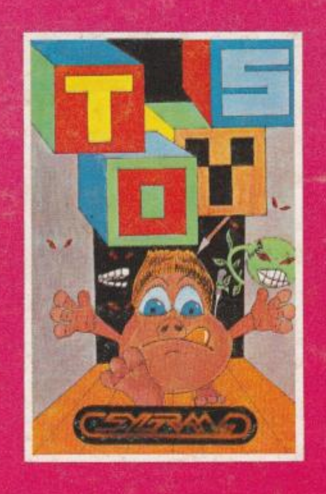
Consultas, dudas, problemas sugerencias, etc. de los lectores que estén relacionados con el sistema AMSTRAD, serán publicadas y respondidas en esta sección.

CLUB

En este apartado daremos cabida a todos los anuncios entre particulares o clubs de informática que estén interesados en cambiar, comprar o vender algo, establecer contactos, etc.

Enviad vuestros mensajes: Mundo AMSTRAD, Tomás López, 3 6.° 28009 MADRID

PARA TU...



AMSTRAD

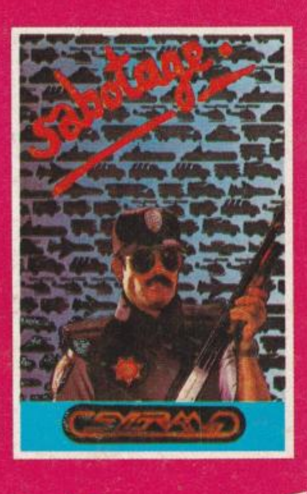
SPECTRUM





MSX

MSX



500 pelas